

A Unified Finite Element Method for Fluid-Structure Interaction

Yongxing Wang*, Peter Jimack, Mark Walkley

School of Computing, University of Leeds, Leeds, UK, LS2 9JT

Abstract

In this article, we present a new unified finite element method (UFEM) for simulation of general Fluid-Structure interaction (FSI) which has the same generality and robustness as monolithic methods but is significantly more computationally efficient and easier to implement. Our proposed approach has similarities with classical immersed finite element methods (IFEMs), by approximating a single velocity and pressure field in the entire domain (i.e. occupied by fluid and solid) on a single mesh, but differs by treating the corrections due to the solid deformation on the left-hand side of the modified fluid flow equations (i.e. implicitly). The method is described in detail, followed by the presentation of multiple computational examples in order to validate it across a wide range of fluid and solid parameters and interactions.

Keywords: Fluid-Structure interaction, Finite element method, Immersed finite element method, Monolithic method, Unified finite element method

1. Introduction

Numerical simulation of fluid-structure interaction is a computational challenge because of its strong nonlinearity, especially when large deformation is considered. Based on how to couple the interaction between fluid and solid, existing numerical methods can be broadly categorized into two approaches: partitioned/segregated methods and monolithic/fully-coupled methods. Similarly, based on how to handle the mesh, they can also be broadly categorized into two further approaches: fitted mesh/conforming methods and unfitted/non-conforming mesh methods [1].

A fitted mesh means that the fluid and solid meshes match each other at the interface, and the nodes on the interface are shared by both the fluid and the solid, which leads to the fact that each interface node has both a fluid velocity and a solid velocity (or displacement) defined on it. It is apparent that

*Corresponding author

Email address: `scywa@leeds.ac.uk` (Yongxing Wang)

the two velocities on each interface node should be consistent. There are typ-
15 ically two methods to handle this: partitioned/segregated methods [2, 3] and
monolithic/fully-coupled methods [4, 5, 6]. The former solve the fluid and solid
equations sequentially and iterate until the velocities become consistent at the
interface. These are more straightforward to implement but can lack robustness
and may fail to converge when there is a significant energy exchange between
20 the fluid and solid [3]. The latter solve the fluid and solid equations simultane-
ously and often use a Lagrange Multiplier to weakly enforce the continuity of
velocity on the interface [6]. This has the advantage of achieving accurate and
stable solutions, however the key computational challenge is to efficiently solve
the large systems of nonlinear algebraic equations arising from the fully-coupled
25 implicit discretization of the fluid and solid equations. Fitted mesh methods can
accurately model wide classes of FSI problems, however maintaining the quality
of the mesh for large solid deformations usually requires a combination of arbi-
trary Lagrangian-Eulerian (ALE) mesh movement and partial or full remeshing
[7]. These add to the computational expense and, when remeshing occurs, can
30 lead to loss of conservation properties of the underlying discretization [8].

Unfitted mesh methods use two meshes to represent the fluid and solid sep-
arately and these do not generally conform to each other on the interface. In
this case, the definition of the fluid problem may be extended to an augmented
domain which includes the solid domain. Similarly to the fitted case, there are
35 also two broad approaches to treat the solid domain: partitioned methods and
monolithic methods. On an unfitted mesh, there is no clear boundary for the
solid problem, so it is not easy to enforce the boundary condition and solve the
solid equation. A wide variety of schemes have been proposed to address this
issue, including the Immersed Finite Element Method (IFEM) [9, 10, 11, 12, 13],
40 the Fictitious Domain (FD) method [14, 15, 16, 17], and the mortar approach
[16, 18]. The IFEM developed from the Immersed Boundary method first intro-
duced by Peskin [19], and has had great success with applications in bioscience
and biomedical fields. The classical IFEM does not solve solid equations at all.
Instead, the solid equations are arranged on the right-hand side of the fluid
45 equations as an FSI force, and these modified fluid equations are solved on the
augmented domain (occupied by fluid and solid). There is also the Modified
IFEM [13], which solves the solid equations explicitly and iterates until conver-
gence. Reference [14] presents a fractional scheme for a rigid body interacting
with the fluid, whilst [15] introduces a fractional step scheme using Distributed
50 Lagrange Multiplier (DLM)/FD for fluid/flexible-body interactions. In the case
of monolithic methods, [16] uses a FD/mortar approach to couple the fluid and
structure, but the coupling is limited to a line (2D) representing the structure.
Reference [18] uses a mortar approach to solve fluid interactions with deformable
and rigid bodies, and [17] also solves a fully-coupled FSI system with hierarchi-
55 cal B-Spline grids. There are also other monolithic methods based on unfitted
meshes [20, 21].

It can be seen that the major methods based on unfitted meshes either avoid
solving the solid equations (IFEM) or solve them with additional variables (two
velocity fields and Lagrange multiplier) in the solid domain. However, physically,

60 there is only one velocity field in the solid domain. In this article, we develop a semi-explicit Unified FEM (UFEM) approach which only solves one velocity variable in the whole/augmented domain. We shall use unfitted meshes to introduce our UFEM, although the methodology can also be applied to fitted meshes.

65 The word “unified” here has two meanings: (1) the equations for fluid and solid are unified in one equation in which only one velocity variable is solved; (2) a range of solid materials, from the very soft to the very hard, may be considered in this one scheme.

The term “semi-explicit” also has two components: (1) we linearize the solid constitutive model (an incompressible neo-Hookean model) explicitly using the value from the last time step; (2) we couple the FSI interaction implicitly by arranging the solid information on the left-hand side of control equations.

The main idea of UFEM is as follows. We first discretize the control equations in time, re-write the solid equation in the form of a fluid equation (using the velocity as a variable rather than the displacement) and re-write the solid constitutive equation in the updated coordinate system. We then combine the fluid and solid equations and discretize them in an augmented domain. Finally the multi-physics problem is solved as a single field.

The UFEM differs from the classical IFEM approach which puts all the solid model information from the last time step explicitly on the right-hand side of the fluid equations. This typically requires the use of a very small time step to simulate the whole FSI system. This IFEM approach works satisfactorily when the solid behaves like a fluid, such as a very soft solid, but can lead to significant errors when the solid behaves quite differently from the fluid, such as a hard solid. The UFEM scheme includes the solid information on the left-hand side and, as we will demonstrate, can simulate a wide range from very soft to very hard solids both accurately and efficiently.

As noted above, monolithic methods strongly couple the fluid and solid models, and discretize them into one implicit nonlinear equation system at each time step. The unknowns include velocity, displacement and a Lagrangian multiplier to enforce consistency of velocity on an interface (fitted mesh) [4, 5, 6] or in a solid domain (unfitted mesh) [16, 17, 18]. One may gain both a stable and an accurate solution from such fully-coupled schemes. However, it is clear that this strategy is very costly, especially for the unfitted mesh case, in which the so called mortar integrals are involved [18]. The UFEM only solves for velocity as unknowns, which is cheaper, but does not lose stability or accuracy as shown by the numerical experiments reported in this paper.

The following sections are organized as follows. In section 2, the control equations and boundary conditions for fluid-structure interactions are introduced; In section 3, the weak form of the FSI system is presented based on the augmented fluid domain. In section 4, details of the linearization of the FSI equations are discussed and the numerical scheme is presented. In section 5, numerical examples are described to validate the proposed UFEM.

2. Governing equations for FSI

In the following context, let

$$(u, v)_\Omega = \int_\Omega u v d\Omega, \quad (1)$$

105 where u and v are functions defined in domain Ω .

110 All subscripts, such as i, j , and k , represent spatial dimension. If they are repeated in one term (including the bracket defined in (1)), it implies summation over the spatial dimension; if they are not repeated, they take the value 1 and 2 for 2D, and 1 to 3 for 3D. All superscripts are used to distinguish fluid and solid (f and s respectively), distinguish different boundaries (Γ^D and Γ^N) or represent time step (n). For example, u_i^f and u_i^s denote the velocity components of fluid and solid respectively, σ_{ij}^f and σ_{ij}^s denote the stress tensor components of fluid and solid respectively, and $(u_i^s)^n$ is a solid velocity component at time t^n .

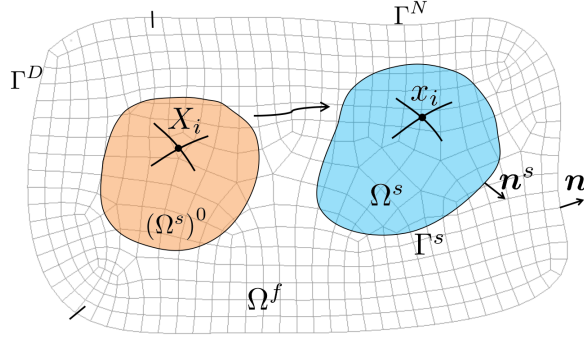


Figure 1: Schematic diagram of FSI, $\Omega = \Omega^f \cup \Omega^s$, $\Gamma = \Gamma^D \cup \Gamma^N$.

In our model we assume an incompressible fluid governed by the following equations in Ω^f as shown in Figure 1:

$$\rho^f \frac{Du_i^f}{Dt} - \frac{\partial \sigma_{ij}^f}{\partial x_j} = \rho^f g_i, \quad (2)$$

$$\frac{\partial u_j^f}{\partial x_j} = 0, \quad (3)$$

$$\sigma_{ij}^f = \mu^f \left(\frac{\partial u_i^f}{\partial x_j} + \frac{\partial u_j^f}{\partial x_i} \right) - p^f \delta_{ij} = \tau_{ij}^f - p^f \delta_{ij}. \quad (4)$$

We also assume an incompressible solid that is governed by the following equations in Ω^s as shown in Figure 1:

$$\rho^s \frac{Du_i^s}{Dt} - \frac{\partial \sigma_{ij}^s}{\partial x_j} = \rho^s g_i, \quad (5)$$

$$\det(\mathbf{F}) = 1, \quad (6)$$

$$\sigma_{ij}^s = \mu^s \left(\frac{\partial x_i^s}{\partial X_k} \frac{\partial x_j^s}{\partial X_k} - \delta_{ij} \right) - p^s \delta_{ij} = \tau_{ij}^s - p^s \delta_{ij}. \quad (7)$$

115 In the above τ_{ij}^f and τ_{ij}^s are the deviatoric stress of the fluid and solid respectively, ρ^f and ρ^s are the density of the fluid and solid respectively, μ^f is the fluid viscosity, and g_i is the acceleration due to gravity. Note that (5)-(7) describe an incompressible neo-Hookean model that is based on [16] and is suitable for large displacements. In this model, μ^s is the shear modulus and p^s is
 120 the pressure of the solid (p^f being the fluid pressure in (4)). We denote by x_i the current coordinates of the solid or fluid, and by X_i the reference coordinates of the solid, whilst $\mathbf{F} = \left[\frac{\partial x_i}{\partial X_j} \right]$ is the deformation tensor of the solid and $\frac{D}{Dt}$ represents the total derivative of time.

On the interface boundary Γ^s :

$$u_i^f = u_i^s, \quad (8)$$

$$\sigma_{ij}^f n_j^s = \sigma_{ij}^s n_j^s, \quad (9)$$

where n_j^s denotes the component of outward pointing unit normal, see Figure 1.

125 Dirichlet and Neumann boundary conditions may be imposed for the fluid:

$$u_i^f = \bar{u}_i \quad \text{on} \quad \Gamma^D, \quad (10)$$

$$\sigma_{ij}^f n_j = \bar{h}_i \quad \text{on} \quad \Gamma^N. \quad (11)$$

Finally, initial conditions are typically set as:

$$u_i^f \Big|_{t=0} = u_i^s \Big|_{t=0} = 0, \quad (12)$$

though they may differ from (12).

Remark 1 Using Jacobi's formula [22]:

$$\frac{d}{dt} \det(\mathbf{F}) = \det(\mathbf{F}) \operatorname{tr} \left(\mathbf{F}^{-1} \frac{d\mathbf{F}}{dt} \right), \quad (13)$$

we have

$$\frac{d}{dt} \det(\mathbf{F}) = \det(\mathbf{F}) \frac{\partial u_j^s}{\partial x_j}, \quad (14)$$

which, using (6), gives

$$\frac{\partial u_j^s}{\partial x_j} = 0. \quad (6')$$

We choose that the reference configuration is the same as the initial configuration, so (6') also implies (6). In our UFEM model, the incompressibility constraint (6') will be used instead of (6).

3. Weak form of FSI equations

In order to obtain a weak formulation we define a combined trial space for velocity as:

$$W = \left\{ (u_i^f, u_i^s) : u_i^f \in H^1(\Omega^f), u_i^s \in H^1(\Omega^s), u_i^s|_{\Gamma^s} = u_i^f|_{\Gamma^s}, u_i^f|_{\Gamma^D} = \bar{u}_i \right\},$$

with a corresponding combined test space for the velocity as:

$$W_0 = \left\{ (v_i^f, v_i^s) : v_i^f \in H^1(\Omega^f), v_i^s \in H^1(\Omega^s), v_i^s|_{\Gamma^s} = v_i^f|_{\Gamma^s}, v_i^f|_{\Gamma^D} = 0 \right\}.$$

Both the trial and test spaces for pressure in Ω^f are $L^2(\Omega^f)$, and both the trial and test spaces for pressure in Ω^s are $L^2(\Omega^s)$. We then perform the following symbolic operations:

$$(\text{Eq.}(2), v_i^f)_{\Omega^f} - (\text{Eq.}(3), q^f)_{\Omega^f} + (\text{Eq.}(5), v_i^s)_{\Omega^s} - (\text{Eq.}(6'), q^s)_{\Omega^s}.$$

Integrating the stress terms by parts, using constitutive equation (4) and (7) and boundary condition (11), the last operations give the following weak form of the FSI system.

Find $(u_i^f, u_i^s) \in W$, $p^f \in L^2(\Omega^f)$ and $p^s \in L^2(\Omega^s)$ such that

$$\begin{aligned} & \rho^f \left(\frac{Du_i^f}{Dt}, v_i^f \right)_{\Omega^f} + \left(\tau_{ij}^f, \frac{\partial v_i^f}{\partial x_j} \right)_{\Omega^f} - \left(p^f, \frac{\partial v_j^f}{\partial x_j} \right)_{\Omega^f} - \left(\frac{\partial u_j^f}{\partial x_j}, q^f \right)_{\Omega^f} \\ & \rho^s \left(\frac{Du_i^s}{Dt}, v_i^s \right)_{\Omega^s} + \left(\tau_{ij}^s, \frac{\partial v_i^s}{\partial x_j} \right)_{\Omega^s} - \left(p^s, \frac{\partial v_j^s}{\partial x_j} \right)_{\Omega^s} - \left(\frac{\partial u_j^s}{\partial x_j}, q^s \right)_{\Omega^s} \\ & = (\bar{h}_i, v_i^f)_{\Gamma^N} + \rho^f (g_i, v_i^f)_{\Omega^f} + \rho^s (g_i, v_i^s)_{\Omega^s}, \end{aligned} \quad (15)$$

$$\forall (v_i^f, v_i^s) \in W_0, \forall q^f \in L^2(\Omega^f) \text{ and } \forall q^s \in L^2(\Omega^s).$$

Note that the integrals on the interface (boundary forces) are also cancelled out using boundary condition (9). This is not surprising because they are internal forces for the whole FSI system considered here.

We next extend the fluid velocity and pressure into solid domain by introducing $u_i = \begin{cases} u_i^f & \text{in } \Omega^f \\ u_i^s & \text{in } \Omega^s \end{cases}$, $v_i = \begin{cases} v_i^f & \text{in } \Omega^f \\ v_i^s & \text{in } \Omega^s \end{cases}$, $p = \begin{cases} p^f & \text{in } \Omega^f \\ p^s & \text{in } \Omega^s \end{cases}$ and $q = \begin{cases} q^f & \text{in } \Omega^f \\ q^s & \text{in } \Omega^s \end{cases}$, then extend the fluid computational domain from Ω^f to an augmented domain Ω , and define a trial space for velocity in Ω as:

$$\bar{W} = \{ u_i : u_i \in H^1(\Omega), R(u_i) = u_i^s \in H^1(\Omega^s), u_i|_{\Gamma^D} = \bar{u}_i \},$$

with a corresponding test space for the velocity as:

$$\bar{W}_0 = \{ v_i : v_i \in H^1(\Omega), R(v_i) = v_i^s \in H^1(\Omega^s), v_i|_{\Gamma^D} = 0 \},$$

where $R(u_i) = u_i|_{\Omega^s}$ is the restriction map.

140

Notice that p^f and p^s are not uniquely determined in (15). In fact, taking $p^f + c$ and $p^s + c$ instead of p^f and p^s respectively, the left-hand side of (15) does not change. This situation can be avoided by fixing the pressure at a selected point (P_0) or by imposing the following constraint [23]:

$$\int_{\Omega^f} p^f d\Omega + \int_{\Omega^s} p^s d\Omega = \int_{\Omega} p d\Omega = 0. \quad (16)$$

We shall use the former approach therefore define the trial space for pressure in Ω as:

$$L_0^2(\Omega) = \{p : p \in L^2(\Omega), p|_{P_0} = 0\}.$$

145 The weak form of the FSI system in the augmented domain Ω can now be reformulated by rearranging equation (15) to yield the following formulation.

Find $u_i \in \overline{W}$ and $p \in L_0^2(\Omega)$ such that

$$\begin{aligned} & \rho^f \left(\frac{Du_i}{Dt}, v_i \right)_{\Omega} + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega} - \left(p, \frac{\partial v_j}{\partial x_j} \right)_{\Omega} - \left(\frac{\partial u_j}{\partial x_j}, q \right)_{\Omega} \\ & + (\rho^s - \rho^f) \left(\frac{Du_i}{Dt}, v_i \right)_{\Omega^s} + \left(\tau_{ij}^s - \tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ & = (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_{\Omega} + (\rho^s - \rho^f) (g_i, v_i)_{\Omega^s}, \end{aligned} \quad (17)$$

$\forall v_i \in \overline{W}_0$ and $\forall q \in L^2(\Omega)$.

Remark 2 The fluid deviatoric stress τ_{ij}^f is generally far smaller than the solid deviatoric stress τ_{ij}^s , so we choose to neglect the fluid deviatoric stress τ_{ij}^f in Ω^s in what follows. Note that the classical IFEM neglects the whole fluid stress σ_{ij}^f when computing the FSI force [9]. An equivalent way of interpreting neglecting τ_{ij}^f in Ω^s is to view the solid as being slightly visco-elastic, having the same viscosity as the fluid.

Remark 3 We treat the solid as a freely moving object in a fluid, so $u_i^s, v_i^s \in H^1(\Omega^s)$ without any boundary constraints in the definition of \overline{W} and \overline{W}_0 respectively. Physically, however, if part of solid boundary is fixed, this fixed boundary can also be regarded as a fixed fluid boundary and implemented as a zero velocity condition in the fluid domain, hence the solid still can be treated as if it were freely moving. Furthermore, the interface boundary condition (8) is automatically built into the solution because we use an augmented solution space \overline{W} which requires $u_i|_{\Omega^s} = u_i^s$.

4. Computational scheme

The integrals in equation (17) are carried out in two different domains as illustrated in Figure 1. We use an Eulerian mesh to represent Ω and an updated

Lagrangian mesh to represent Ω^s , therefore the total time derivatives in these two different domains have different expressions, i.e:

$$\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \quad \text{in } \Omega, \quad (18)$$

and

$$\frac{Du_i^s}{Dt} = \frac{\partial u_i^s}{\partial t} \quad \text{in } \Omega^s. \quad (19)$$

Standard FEM isoparametric interpolation may be used to transfer data between the two meshes. Firstly, based on the above two equations (18) and (19), we discretize (17) in time using a backward finite difference. Then omitting the superscript $n + 1$, showing the solution is at the end of the time step, for convenience, we obtain:

$$\begin{aligned} & \rho^f \left(\frac{u_i - u_i^n}{\Delta t} + u_j \frac{\partial u_i}{\partial x_j}, v_i \right)_{\Omega} + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega} - \left(p, \frac{\partial v_j}{\partial x_j} \right)_{\Omega} - \left(\frac{\partial u_j}{\partial x_j}, q \right)_{\Omega} \\ & + (\rho^s - \rho^f) \left(\frac{u_i - u_i^n}{\Delta t}, v_i \right)_{\Omega^s} + \left(\tau_{ij}^s, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ & = (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_{\Omega} + (\rho^s - \rho^f) (g_i, v_i)_{\Omega^s}. \end{aligned} \quad (20)$$

Using the splitting method of [24, Chapter 3], equation (20) can be expressed in the following two steps.

(1) Convection step:

$$\rho^f \left(\frac{u_i^* - u_i^n}{\Delta t} + u_j^* \frac{\partial u_i^*}{\partial x_j}, v_i \right)_{\Omega} = 0; \quad (21)$$

(2) Diffusion step:

$$\begin{aligned} & \rho^f \left(\frac{u_i - u_i^*}{\Delta t}, v_i \right)_{\Omega} + \left(\tau_{ij}^f, \frac{\partial v_i}{\partial x_j} \right)_{\Omega} - \left(p, \frac{\partial v_j}{\partial x_j} \right)_{\Omega} - \left(\frac{\partial u_j}{\partial x_j}, q \right)_{\Omega} \\ & + (\rho^s - \rho^f) \left(\frac{u_i - u_i^*}{\Delta t}, v_i \right)_{\Omega^s} + \left(\tau_{ij}^s, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ & = (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_{\Omega} + (\rho^s - \rho^f) (g_i, v_i)_{\Omega^s}. \end{aligned} \quad (22)$$

165 The treatment of the above two steps is described separately in the following subsections.

4.1. Linearization of the convection step

170 In this section, two methods are introduced to treat the convection equation: the implicit Least-squares method and the explicit Taylor-Galerkin method, both of which can be used in the framework of our UFEM scheme. Some numerical results for comparison between these two methods are discussed subsequently in section 5.

4.1.1. Implicit Least-squares method

It is possible to linearize (21) using the value of u_i from the last time step:

$$u_j^* \frac{\partial u_i^*}{\partial x_j} \approx u_j^* \frac{\partial u_i^n}{\partial x_j} + u_j^n \frac{\partial u_i^*}{\partial x_j} - u_j^n \frac{\partial u_i^n}{\partial x_j}. \quad (23)$$

Substituting (23) into equation (21) gives,

$$\left(u_i^* + \Delta t \left(u_j^* \frac{\partial u_i^n}{\partial x_j} + u_j^n \frac{\partial u_i^*}{\partial x_j} \right), v_i \right)_{\Omega} = \left(u_i^n + \Delta t u_j^n \frac{\partial u_i^n}{\partial x_j}, v_i \right)_{\Omega}. \quad (24)$$

For the Least-squares method [25], we may choose the test function in the following form:

$$v_i = L(w_i) = w_i + \Delta t \left(w_j \frac{\partial u_i^n}{\partial x_j} + u_j^n \frac{\partial w_i}{\partial x_j} \right), \quad (25)$$

where $w_i \in \overline{W}_0$. In such a case, the weak form of (21) is:

$$(L(u_i^*), L(w_i))_{\Omega} = \left(u_i^n + \Delta t u_j^n \frac{\partial u_i^n}{\partial x_j}, L(w_i) \right)_{\Omega}. \quad (26)$$

In our UFEM a standard biquadratic finite element space is used to discretize equation (26) directly, although other spaces could be used.

4.1.2. Explicit Taylor-Galerkin method

It is also possible to linearize equation (21) as:

$$\left(\frac{u_i^* - u_i^n}{\Delta t} + \frac{1}{2} u_j^n \frac{\partial}{\partial x_j} (u_i^* + u_i^n), v_i \right)_{\Omega} = 0, \quad (27)$$

or

$$\left(\frac{u_i^* - u_i^n}{\Delta t} + u_j^n \frac{\partial u_i^n}{\partial x_j}, v_i \right)_{\Omega} = 0. \quad (28)$$

Re-write (28) as:

$$u_i^* = u_i^n - \Delta t u_j^n \frac{\partial u_i^n}{\partial x_j}, \quad (29)$$

and substitute (29) into equation (27), we have

$$\left(\frac{u_i^* - u_i^n}{\Delta t} + u_j^n \frac{\partial u_i^n}{\partial x_j} - \frac{\Delta t}{2} u_j^n \frac{\partial}{\partial x_j} \left(u_k^n \frac{\partial u_i^n}{\partial x_k} \right), v_i \right)_{\Omega} = 0. \quad (30)$$

Notice that a second order derivative exists in the last equation. In practice, one does not need to calculate the second order derivative, instead, Integration by parts may be used to reduce the order:

$$\left(\frac{\partial}{\partial x_j} \left(u_k \frac{\partial u_i}{\partial x_k} \right), v_i \right)_{\Omega} = \left(u_k \frac{\partial u_i}{\partial x_k}, v_i \right)_{\Gamma^N} - \left(u_k \frac{\partial u_i}{\partial x_k}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega}. \quad (31)$$

The boundary integral in the last equation can be neglected if u_i is the solution of the previous diffusion step, which means no convection exists on the boundary after the diffusion step. Using (31), equation (30) may be approximated as:

$$\left(\frac{u_i^* - u_i^n}{\Delta t} + u_j^n \frac{\partial u_i^n}{\partial x_j}, v_i \right)_\Omega = -\frac{\Delta t}{2} \left(u_k^n \frac{\partial u_i^n}{\partial x_k}, u_j^n \frac{\partial v_i}{\partial x_j} \right)_\Omega. \quad (32)$$

At last the weak form of the Taylor-Galerkin method [24, Chapter 2] can be expressed, by rearranging the last equation, as:

$$(u_i^*, v_i)_\Omega = \left(u_i^n - \Delta t u_j^n \frac{\partial u_i^n}{\partial x_j}, v_i \right)_\Omega - \frac{\Delta t^2}{2} \left(u_k^n \frac{\partial u_i^n}{\partial x_k}, u_j^n \frac{\partial v_i}{\partial x_j} \right)_\Omega. \quad (33)$$

This Taylor-Galerkin method is explicit, however a small time step is usually needed to keep the scheme stable.

4.2. Linearization of the diffusion step

In both the above and the following context, the derivative $\frac{\partial}{\partial x_i}$ on the updated solid mesh is computed at the current known coordinates x_i^n , that is to say $\frac{\partial}{\partial x_i} = \frac{\partial}{\partial x_i^n}$. Furthermore, τ_{ij}^s in equations (22), has a nonlinear relationship with x_i , i.e.:

$$\tau_{ij}^s = (\tau_{ij}^s)^{n+1} = \mu^s \left(\frac{\partial x_i^{n+1}}{\partial X_k} \frac{\partial x_j^{n+1}}{\partial X_k} - \delta_{ij} \right). \quad (34)$$

Using a chain rule, the last equation can also be expressed as:

$$\begin{aligned} (\tau_{ij}^s)^{n+1} &= \mu^s \frac{\partial x_i^{n+1}}{\partial x_k^n} \frac{\partial x_k^n}{\partial X_m} \frac{\partial x_l^n}{\partial X_m} \frac{\partial x_j^{n+1}}{\partial x_l^n} - \mu^s \delta_{ij} \\ &+ \mu^s \frac{\partial x_i^{n+1}}{\partial x_k^n} \frac{\partial x_j^{n+1}}{\partial x_k^n} - \mu^s \frac{\partial x_i^{n+1}}{\partial x_k^n} \delta_{kl} \frac{\partial x_j^{n+1}}{\partial x_l^n} \end{aligned} \quad (35)$$

or

$$\begin{aligned} (\tau_{ij}^s)^{n+1} &= \mu^s \left(\frac{\partial x_i^{n+1}}{\partial x_k^n} \frac{\partial x_j^{n+1}}{\partial x_k^n} - \delta_{ij} \right) \\ &+ \mu^s \frac{\partial x_i^{n+1}}{\partial x_k^n} \left(\frac{\partial x_k^n}{\partial X_m} \frac{\partial x_l^n}{\partial X_m} - \delta_{kl} \right) \frac{\partial x_j^{n+1}}{\partial x_l^n}, \end{aligned} \quad (36)$$

and then $(\tau_{ij}^s)^{n+1}$ can be expressed by the current coordinate x_i^n as follows:

$$(\tau_{ij}^s)^{n+1} = \mu^s \left(\frac{\partial x_i^{n+1}}{\partial x_k^n} \frac{\partial x_j^{n+1}}{\partial x_k^n} - \delta_{ij} \right) + \frac{\partial x_i^{n+1}}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial x_j^{n+1}}{\partial x_l^n}. \quad (37)$$

Using $x_i^{n+1} - x_i^n = u_i^{n+1} \Delta t$ which is the displacement at the current step, the last equation can also be expressed as:

$$\begin{aligned} (\tau_{ij}^s)^{n+1} &= \mu^s \Delta t \left(\frac{\partial u_i^{n+1}}{\partial x_j^n} + \frac{\partial u_j^{n+1}}{\partial x_i^n} + \Delta t \frac{\partial u_i^{n+1}}{\partial x_k^n} \frac{\partial u_j^{n+1}}{\partial x_k^n} \right) + (\tau_{ij}^s)^n \\ &+ \Delta t^2 \frac{\partial u_i^{n+1}}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial u_j^{n+1}}{\partial x_l^n} + \Delta t \frac{\partial u_i^{n+1}}{\partial x_k^n} (\tau_{kj}^s)^n + \Delta t (\tau_{il}^s)^n \frac{\partial u_j^{n+1}}{\partial x_l^n}. \end{aligned} \quad (38)$$

There are two nonlinear terms in the last equation. Using a Newton method, they can be linearized as follows.

$$\frac{\partial u_i^{n+1}}{\partial x_k^n} \frac{\partial u_j^{n+1}}{\partial x_k^n} = \frac{\partial u_i^{n+1}}{\partial x_k^n} \frac{\partial u_j^n}{\partial x_k^n} + \frac{\partial u_i^n}{\partial x_k^n} \frac{\partial u_j^{n+1}}{\partial x_k^n} - \frac{\partial u_i^n}{\partial x_k^n} \frac{\partial u_j^n}{\partial x_k^n} \quad (39)$$

and

$$\begin{aligned} \frac{\partial u_i^{n+1}}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial u_j^{n+1}}{\partial x_l^n} &= \frac{\partial u_i^{n+1}}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l^n} \\ &+ \frac{\partial u_i^n}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial u_j^{n+1}}{\partial x_l^n} - \frac{\partial u_i^n}{\partial x_k^n} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l^n}. \end{aligned} \quad (40)$$

Substituting (38)-(40) into (22) and dropping off the superscripts $n+1$ of u_i^{n+1} for notation convenience, this may be expressed as:

$$\begin{aligned} &\rho^f \left(\frac{u_i - u_i^*}{\Delta t}, v_i \right)_{\Omega} + (\rho^s - \rho^f) \left(\frac{u_i - (u_i^s)^n}{\Delta t}, v_i \right)_{\Omega^s} \\ &+ \mu^f \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega} - \left(p, \frac{\partial v_j}{\partial x_j} \right)_{\Omega} - \left(\frac{\partial u_j}{\partial x_j}, q \right)_{\Omega} \\ &+ \mu^s \Delta t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \Delta t \frac{\partial u_i}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j}{\partial x_k}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ &+ \Delta t^2 \left(\frac{\partial u_i}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} + \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j}{\partial x_l}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ &+ \Delta t \left(\frac{\partial u_i}{\partial x_k} (\tau_{kj}^s)^n + (\tau_{il}^s)^n \frac{\partial u_j}{\partial x_l}, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s} \\ &= (\bar{h}_i, v_i)_{\Gamma^N} + \rho^f (g_i, v_i)_{\Omega} + (\rho^s - \rho^f) (g_i, v_i)_{\Omega^s} \\ &+ \left(\mu^s \Delta t^2 \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{ij}^s)^n, \frac{\partial v_i}{\partial x_j} \right)_{\Omega^s}. \end{aligned} \quad (41)$$

180 The spatial discretization of the above linearized weak form will be discussed in the following section.

4.3. Discretization in space

In the 2D case, which is considered in the remainder of this paper, a standard Taylor-Hood element Q2Q1 (9-node biquadratic quadrilateral for velocity and

185 4-node bilinear quadrilateral for pressure) is used to discretize in space. We first
 discretize the domain Ω to get Ω^h , then define finite dimensional subspaces of
 \bar{W} and \bar{W}_0 as follows.

The solution space for each component of velocity:

$$\bar{W}^h = \{u_i^h : u_i^h \in H^{1h}(\Omega^h), R^h(u_i^h) = u_i^{sh}, u_i^h|_{\Gamma^D} = \bar{u}_i^h\},$$

whilst test space for velocity is

$$\bar{W}_0^h = \{v_i^h : v_i^h \in H^{1h}(\Omega^h), R^h(v_i^h) = v_i^{sh}, v_i^h|_{\Gamma^D} = 0\}.$$

We also discretize the domain Ω^s to get Ω^{sh} , and both the discretized trial
 space and test space on the solid domain are $H^{1h}(\Omega^{sh})$ based on the discussion
 of Remark 3.

190 The solution and test spaces for pressure are $L_0^{2h}(\Omega^h)$ and $L^{2h}(\Omega^h)$ respec-
 tively, which represent the finite dimensional subspaces of $L_0^2(\Omega)$ and $L^2(\Omega)$, re-
 spectively, based on continuous piecewise bilinear functions. $H^{1h}(\Omega^h)$ ($H^{1h}(\Omega^{sh})$)
 represents the finite dimensional subspace of $H^1(\Omega)$ ($H^1(\Omega^s)$) based upon con-
 tinuous piecewise biquadratic functions. Then equation (41) can be discretized
 as:

$$\begin{aligned} & \rho^f \left(\frac{u_i^h - u_i^{*h}}{\Delta t}, v_i^h \right)_{\Omega^h} + (\rho^s - \rho^f) \left(\frac{u_i^{sh} - (u_i^{sh})^n}{\Delta t}, v_i^{sh} \right)_{\Omega^{sh}} \\ & + \mu^f \left(\frac{\partial u_i^h}{\partial x_j} + \frac{\partial u_j^h}{\partial x_i}, \frac{\partial v_i^h}{\partial x_j} \right)_{\Omega^h} - \left(p^h, \frac{\partial v_j^h}{\partial x_j} \right)_{\Omega^h} - \left(\frac{\partial u_j^h}{\partial x_j}, q^h \right)_{\Omega^h} \\ & + \mu^s \Delta t \left(\frac{\partial u_i^{sh}}{\partial x_j} + \frac{\partial u_j^{sh}}{\partial x_i} + \Delta t \frac{\partial u_i^{sh}}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j^{sh}}{\partial x_k}, \frac{\partial v_i^{sh}}{\partial x_j} \right)_{\Omega^{sh}} \\ & + \Delta t^2 \left(\frac{\partial u_i^{sh}}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} + \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^{sh}}{\partial x_l}, \frac{\partial v_i^{sh}}{\partial x_j} \right)_{\Omega^{sh}} \\ & + \Delta t \left(\frac{\partial u_i^{sh}}{\partial x_k} (\tau_{kj}^s)^n + (\tau_{il}^s)^n \frac{\partial u_j^{sh}}{\partial x_l}, \frac{\partial v_i^{sh}}{\partial x_j} \right)_{\Omega^{sh}} \\ & = (\bar{h}_i, v_i^h)_{\Gamma^{Nh}} + \rho^f (g_i, v_i^h)_{\Omega^h} + (\rho^s - \rho^f) (g_i, v_i^{sh})_{\Omega^{sh}} \\ & + \left(\mu^s \Delta t^2 \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{ij}^s)^n, \frac{\partial v_i^{sh}}{\partial x_j} \right)_{\Omega^{sh}}. \end{aligned} \quad (42)$$

Notice that in the continuous space \bar{W} , we have the restriction map $R(u_i) = u_i|_{\Omega^s} = u_i^s$, while in the discretized space \bar{W}^h , we use the standard FEM isopara-
 metric transformation R^h to represent the map, i.e.

$$u_i^{sh} = R^h(u_i^h), \quad (43)$$

where subscript i denotes the velocity components in each space dimension.

Let $\tilde{\mathbf{u}}_i = (\tilde{u}_{i1}, \tilde{u}_{i2} \cdots \tilde{u}_{iN^f})^T$ and $\tilde{\mathbf{u}}_i^s = (\tilde{u}_{i1}^s, \tilde{u}_{i2}^s \cdots \tilde{u}_{iN^s}^s)^T$ denote the i^{th} components of the nodal velocity vectors on the fluid and solid meshes respectively, and $\boldsymbol{\varphi} = (\varphi_1, \varphi_2 \cdots \varphi_{N^f})^T$ and $\boldsymbol{\varphi}^s = (\varphi_1^s, \varphi_2^s \cdots \varphi_{N^s}^s)^T$ denote the vector of velocity basis functions on the fluid and solid meshes respectively, where N^f and N^s are the number of nodes of fluid and solid mesh respectively. Then equation (43) can be expressed as:

$$\tilde{u}_{ik}^s \varphi_k^s = R^h(\tilde{u}_{ik} \varphi_k). \quad (44)$$

The FEM isoparametric transformation defines R^h from $\tilde{\mathbf{u}}_i$ to $\tilde{\mathbf{u}}_i^s$ as follows:

$$\tilde{u}_{ik}^s = R^h(\tilde{u}_{il}) = \tilde{u}_{il} R_{lk}, \quad (45)$$

where $R_{lk} = \varphi_l(\mathbf{x}_k)$, $\mathbf{x}_k (k = 1, 2 \cdots N^s)$ is the current coordinate of the k^{th} node on the solid mesh. Therefore,

$$u_i^{sh} = \tilde{u}_{ik}^s \varphi_k^s = \tilde{u}_{il} R_{lk} \varphi_k^s. \quad (46)$$

For velocity test functions, we similarly have

$$v_i^{sh} = \tilde{v}_{ik}^s \varphi_k^s = \tilde{v}_{il} R_{lk} \varphi_k^s, \quad (47)$$

where $\tilde{\mathbf{v}}_i = (\tilde{v}_{i1}, \tilde{v}_{i2} \cdots \tilde{v}_{iN^f})^T$ is an arbitrary nodal velocity (virtual velocity) vector on the fluid mesh, which satisfies the homogeneous Dirichlet boundary condition.

On the fluid mesh, velocity and pressure can also be expressed as follows:

$$u_i^h = \tilde{u}_{ik} \varphi_k, \quad (48)$$

$$v_i^h = \tilde{v}_{ik} \varphi_k, \quad (49)$$

$$p^h = \tilde{p}_k \psi_k, \quad (50)$$

$$q^h = \tilde{q}_k \psi_k, \quad (51)$$

where $\boldsymbol{\psi} = (\psi_1, \psi_2 \cdots \psi_{N^p})^T$ is the vector of pressure basis functions, $\tilde{\mathbf{p}} = (\tilde{p}_1, \tilde{p}_2, \cdots \tilde{p}_{N^p})^T$ is the nodal pressure vector, and $\tilde{\mathbf{q}} = (\tilde{q}_1, \tilde{q}_2, \cdots \tilde{q}_{N^p})^T$ is an arbitrary nodal pressure vector. N^p denotes the number of nodes on the fluid mesh at which only pressure is defined.

Substituting (46)-(51) into (42), we have

$$\begin{aligned}
& \rho^f \left(\frac{\tilde{u}_{ik} - \tilde{u}_{ik}^*}{\Delta t} \varphi_k, \tilde{v}_{im} \varphi_m \right)_{\Omega^h} \\
& + (\rho^s - \rho^f) \left(\frac{\tilde{u}_{il} - \tilde{u}_{il}^n}{\Delta t} R_{lk} \varphi_k^s, \tilde{v}_{ir} R_{rm} \varphi_m^s \right)_{\Omega^{sh}} \\
& + \mu^f \left(\tilde{u}_{ik} \frac{\partial \varphi_k}{\partial x_j} + \tilde{u}_{jk} \frac{\partial \varphi_k}{\partial x_i}, \tilde{v}_{im} \frac{\partial \varphi_m}{\partial x_j} \right)_{\Omega^h} \\
& - \left(\tilde{p}_k \psi_k, \tilde{v}_{jm} \frac{\partial \varphi_m}{\partial x_j} \right)_{\Omega^h} - \left(\tilde{u}_{jk} \frac{\partial \varphi_k}{\partial x_j}, \tilde{q}_m \psi_m \right)_{\Omega^h} \\
& + \mu^s \Delta t \left(\tilde{u}_{il} R_{lk} \frac{\partial \varphi_k^s}{\partial x_j} + \tilde{u}_{jl} R_{lk} \frac{\partial \varphi_k^s}{\partial x_i}, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \\
& + \mu^s \Delta t^2 \left(\tilde{u}_{ia} R_{ab} \frac{\partial \varphi_b^s}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \tilde{u}_{ja} R_{ab} \frac{\partial u_i^n}{\partial x_k} \frac{\partial \varphi_b^s}{\partial x_k}, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \\
& + \Delta t^2 \left(\tilde{u}_{ia} R_{ab} \frac{\partial \varphi_b^s}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l}, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \\
& + \Delta t^2 \left(\tilde{u}_{ja} R_{ab} \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial \varphi_b^s}{\partial x_l}, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \\
& + \Delta t \left(\tilde{u}_{ia} R_{ab} \frac{\partial \varphi_b^s}{\partial x_k} (\tau_{kj}^s)^n + (\tau_{il}^s)^n \tilde{u}_{ja} R_{ab} \frac{\partial \varphi_b^s}{\partial x_l}, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \\
& = (\bar{h}_i, \tilde{v}_{im} \varphi_m)_{\Gamma^{Nh}} + \rho^f (g_i, \tilde{v}_{im} \varphi_m)_{\Omega^h} + (\rho^s - \rho^f) (g_i, \tilde{v}_{ir} R_{rm} \varphi_m^s)_{\Omega^{sh}} \\
& + \left(\mu^s \Delta t^2 \frac{\partial u_i^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_i^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{ij}^s)^n, \tilde{v}_{ir} R_{rm} \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}}.
\end{aligned} \tag{52}$$

Let $\tilde{\mathbf{u}} = (\tilde{\mathbf{u}}_1^T, \tilde{\mathbf{u}}_2^T)^T$ and $\tilde{\mathbf{v}} = (\tilde{\mathbf{v}}_1^T, \tilde{\mathbf{v}}_2^T)^T$, we then express (52) in the following matrix form:

$$\begin{aligned}
& \tilde{\mathbf{v}}^T \mathbf{M} \frac{\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^*}{\Delta t} + \tilde{\mathbf{v}}^T \mathbf{D}^T \mathbf{M}^s \mathbf{D} \frac{\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^n}{\Delta t} \\
& + \tilde{\mathbf{v}}^T \mathbf{K} \tilde{\mathbf{u}} + \tilde{\mathbf{v}}^T \mathbf{B} \tilde{\mathbf{p}} + \tilde{\mathbf{q}}^T \mathbf{B}^T \tilde{\mathbf{u}} + \tilde{\mathbf{v}}^T \mathbf{D}^T \mathbf{K}^s \mathbf{D} \tilde{\mathbf{u}} \\
& = \tilde{\mathbf{v}}^T \mathbf{f} + \tilde{\mathbf{v}}^T \mathbf{D}^T \mathbf{f}^s,
\end{aligned} \tag{53}$$

or

$$(\tilde{\mathbf{v}}^T, \tilde{\mathbf{q}}^T) \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{pmatrix} = (\tilde{\mathbf{v}}^T, \tilde{\mathbf{q}}^T) \begin{pmatrix} \tilde{\mathbf{b}} \\ \mathbf{0} \end{pmatrix}, \tag{54}$$

where

$$\mathbf{A} = \mathbf{M}/\Delta t + \mathbf{K} + \mathbf{D}^T (\mathbf{M}^s/\Delta t + \mathbf{K}^s) \mathbf{D}$$

and

$$\mathbf{b} = \mathbf{f} + \mathbf{D}^T \mathbf{f}^s + \mathbf{M} \tilde{\mathbf{u}}^*/\Delta t + \mathbf{D}^T \mathbf{M}^s \mathbf{D} \tilde{\mathbf{u}}^n/\Delta t.$$

The matrix

$$\mathbf{M} = \rho^f \begin{bmatrix} \mathbf{M}_{11} & \\ & \mathbf{M}_{22} \end{bmatrix} \tag{55}$$

is the velocity mass matrix of the fluid, where

$$(\mathbf{M}_{11})_{km} = (\mathbf{M}_{22})_{km} = (\varphi_k, \varphi_m)_{\Omega^h}, (k, m = 1, 2, \dots, N^f).$$

The matrix

$$\mathbf{M}^s = (\rho^s - \rho^f) \begin{bmatrix} \mathbf{M}_{11}^s & \\ & \mathbf{M}_{22}^s \end{bmatrix} \quad (56)$$

is the velocity mass matrix of the solid, where

$$(\mathbf{M}_{11}^s)_{km} = (\mathbf{M}_{22}^s)_{km} = (\varphi_k^s, \varphi_m^s)_{\Omega^{sh}}, (k, m = 1, 2, \dots, N^s).$$

\mathbf{K} is the stiffness matrix of the fluid:

$$\mathbf{K} = \mu^f \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}, \quad (57)$$

where

$$\begin{aligned} (\mathbf{K}_{11})_{km} &= 2 \left(\frac{\partial \varphi_k}{\partial x_1}, \frac{\partial \varphi_m}{\partial x_1} \right)_{\Omega^h} + \left(\frac{\partial \varphi_k}{\partial x_2}, \frac{\partial \varphi_m}{\partial x_2} \right)_{\Omega^h}, \\ (\mathbf{K}_{22})_{km} &= 2 \left(\frac{\partial \varphi_k}{\partial x_2}, \frac{\partial \varphi_m}{\partial x_2} \right)_{\Omega^h} + \left(\frac{\partial \varphi_k}{\partial x_1}, \frac{\partial \varphi_m}{\partial x_1} \right)_{\Omega^h}, \\ (\mathbf{K}_{12})_{km} &= \left(\frac{\partial \varphi_k}{\partial x_1}, \frac{\partial \varphi_m}{\partial x_2} \right)_{\Omega^h}, \\ (\mathbf{K}_{21})_{km} &= (\mathbf{K}_{12})_{mk} = \left(\frac{\partial \varphi_k}{\partial x_2}, \frac{\partial \varphi_m}{\partial x_1} \right)_{\Omega^h}, \end{aligned}$$

and $k, m = 1, 2, \dots, N^f$.

\mathbf{K}^s is the stiffness matrix of the solid:

$$\mathbf{K}^s = \begin{bmatrix} \mathbf{K}_{11}^s & \mathbf{K}_{12}^s \\ \mathbf{K}_{21}^s & \mathbf{K}_{22}^s \end{bmatrix}, \quad (58)$$

where

$$\begin{aligned} (\mathbf{K}_{11}^s)_{bm} &= \mu^s \Delta t^2 \left(\frac{\partial \varphi_b^s}{\partial x_1}, \frac{\partial \varphi_m^s}{\partial x_1} \right)_{\Omega^{sh}} + \mu^s \Delta t \left(\frac{\partial \varphi_b^s}{\partial x_2}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} \\ &+ 2\mu^s \Delta t^2 \left(\frac{\partial \varphi_b^s}{\partial x_k} \frac{\partial u_1^n}{\partial x_k}, \frac{\partial \varphi_m^s}{\partial x_1} \right)_{\Omega^{sh}} + \mu^s \Delta t^2 \left(\frac{\partial \varphi_b^s}{\partial x_k} \frac{\partial u_2^n}{\partial x_k}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} \\ &+ 2\Delta t^2 \left(\frac{\partial \varphi_b^s}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_1^n}{\partial x_l}, \frac{\partial \varphi_m^s}{\partial x_1} \right)_{\Omega^{sh}} + \Delta t^2 \left(\frac{\partial \varphi_b^s}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_2^n}{\partial x_l}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} \\ &+ 2\Delta t \left(\frac{\partial \varphi_b^s}{\partial x_k} (\tau_{k1}^s)^n, \frac{\partial \varphi_m^s}{\partial x_1} \right)_{\Omega^{sh}} + \Delta t \left(\frac{\partial \varphi_b^s}{\partial x_k} (\tau_{k2}^s)^n, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}}. \end{aligned}$$

It can be seen from the pattern of the above matrices that one can get \mathbf{K}_{22}^s by changing the subscript 1 to 2, and changing 2 to 1 in the formula of \mathbf{K}_{11}^s .

Similarly, the elements of \mathbf{K}_{12}^s can be expressed as:

$$\begin{aligned} (\mathbf{K}_{12}^s)_{bm} &= \mu^s \Delta t \left(\frac{\partial \varphi_b^s}{\partial x_1}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} + \mu^s \Delta t^2 \left(\frac{\partial u_1^n}{\partial x_k} \frac{\partial \varphi_b^s}{\partial x_k}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} \\ &+ \Delta t^2 \left(\frac{\partial u_1^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial \varphi_b^s}{\partial x_l}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}} + \Delta t \left((\tau_{1k}^s)^n \frac{\partial \varphi_b^s}{\partial x_k}, \frac{\partial \varphi_m^s}{\partial x_2} \right)_{\Omega^{sh}}, \end{aligned}$$

and $(\mathbf{K}_{21}^s)_{bm} = (\mathbf{K}_{12}^s)_{mb}$, $(b, m = 1, 2, \dots, N^s)$.
The matrix \mathbf{B} has the following expression.

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}, \quad (59)$$

where

$$(\mathbf{B}_1)_{mk} = \left(\psi_k, \frac{\partial \varphi_m}{\partial x_1} \right)_{\Omega^h}, (\mathbf{B}_2)_{mk} = \left(\psi_k, \frac{\partial \varphi_m}{\partial x_2} \right)_{\Omega^h}$$

$(k = 1, 2, \dots, N^p \text{ and } m = 1, 2, \dots, N^f)$. The vector

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix} \quad (60)$$

is the fluid force vector, where

$$(\mathbf{f}_1)_m = \rho^f (g_1, \varphi_m)_{\Omega^h} + (\bar{h}_1, \varphi_m)_{\Gamma^{Nh}},$$

and

$$(\mathbf{f}_2)_m = \rho^f (g_2, \varphi_m)_{\Omega^h} + (\bar{h}_2, \varphi_m)_{\Gamma^{Nh}}$$

$(m = 1, 2, \dots, N^f)$. The vector

$$\mathbf{f}^s = \begin{pmatrix} \mathbf{f}_1^s \\ \mathbf{f}_2^s \end{pmatrix} \quad (61)$$

is the solid force vector, where

$$\begin{aligned} (\mathbf{f}_1^s)_m &= (\rho^s - \rho^f) (g_1, \varphi_m^s)_{\Omega^{sh}} \\ &+ \left(\mu^s \Delta t^2 \frac{\partial u_1^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_1^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{1j}^s)^n, \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \end{aligned}$$

and

$$\begin{aligned} (\mathbf{f}_2^s)_m &= (\rho^s - \rho^f) (g_2, \varphi_m^s)_{\Omega^{sh}} \\ &+ \left(\mu^s \Delta t^2 \frac{\partial u_2^n}{\partial x_k} \frac{\partial u_j^n}{\partial x_k} + \Delta t^2 \frac{\partial u_2^n}{\partial x_k} (\tau_{kl}^s)^n \frac{\partial u_j^n}{\partial x_l} - (\tau_{2j}^s)^n, \frac{\partial \varphi_m^s}{\partial x_j} \right)_{\Omega^{sh}} \end{aligned}$$

$(m = 1, 2, \dots, N^s)$. Finally, matrix \mathbf{D} is the FEM interpolation matrix which can be expressed as:

$$\mathbf{D} = \begin{bmatrix} \mathbf{R}^T \\ \mathbf{R}^T \end{bmatrix}. \quad (62)$$

Using the arbitrariness of our test vectors $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{q}}$, one can obtain the following linear algebraic equation for the whole FSI system from equation (54):

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{b}} \\ \mathbf{0} \end{pmatrix}. \quad (63)$$

4.4. The UFEM algorithm

200 Having derived a discrete system of equations we now describe the solution algorithm at each time step.

- (1) Given the solid configuration $(\mathbf{x}^s)^n$ and velocity field $\tilde{\mathbf{u}}^n = \begin{cases} (\tilde{\mathbf{u}}^f)^n & \text{in } \Omega^f \\ (\tilde{\mathbf{u}}^s)^n & \text{in } \Omega^s \end{cases}$ at time step n .
- (2) Discretize the convection equation (26) or (33) and solve it to get an intermediate velocity \mathbf{u}^* .
- 205 (3) Compute the interpolation matrix and solve equation (63) using \mathbf{u}^* and $(\tilde{\mathbf{u}}^s)^n$ as initial values to get velocity field $\tilde{\mathbf{u}}^{n+1}$.
- (4) Compute solid velocity $(\tilde{\mathbf{u}}^s)^{n+1} = \mathbf{D}\tilde{\mathbf{u}}^{n+1}$ and update the solid mesh by $(\mathbf{x}^s)^{n+1} = (\mathbf{x}^s)^n + \Delta t (\tilde{\mathbf{u}}^s)^{n+1}$, then go to step (1) for the next time step.

210 **Remark 4** When implementing the UFEM algorithm, it is unnecessary to perform the matrix multiplication $\mathbf{D}^T \mathbf{K}^s \mathbf{D}$ in (53) globally, because the FEM interpolation is locally based. All the matrix operations can be computed based on the local element matrices only. Alternatively, if an iterative solver is used, it is actually unnecessary to compute $\mathbf{D}^T \mathbf{K}^s \mathbf{D}$. What an iterative step needs is to compute $(\mathbf{D}^T \mathbf{K}^s \mathbf{D}) \mathbf{u}$ for a given vector \mathbf{u} , therefore one can compute $\mathbf{D}\mathbf{u}$ first, then $\mathbf{K}^s (\mathbf{D}\mathbf{u})$, and last $\mathbf{D}^T (\mathbf{K}^s \mathbf{D}\mathbf{u})$.

5. Numerical experiments

220 In this section, we present some numerical examples that have been selected to allow us to assess our proposed UFEM. We shall demonstrate the convergence of UFEM in time and space, and compare results obtained by the UFEM with those obtained using monolithic approaches and IFEM, as well as compare against results from laboratory experiment.

225 In order to improve the computational efficiency, an adaptive spatial mesh with hanging nodes is used in all the following numerical experiments. Readers can reference Appendix A for details of the treatment of hanging nodes.

5.1. Oscillation of a flexible leaflet oriented across the flow direction

230 This numerical example is used by [15, 16, 17] to validate their methods. We first use the same parameters as used in the above three publications in order to compare results and test convergence in time and space, then use a range of parameters to show the robustness of our UFEM. The implicit Least-squares method is used to treat the convection step in all these tests unless otherwise

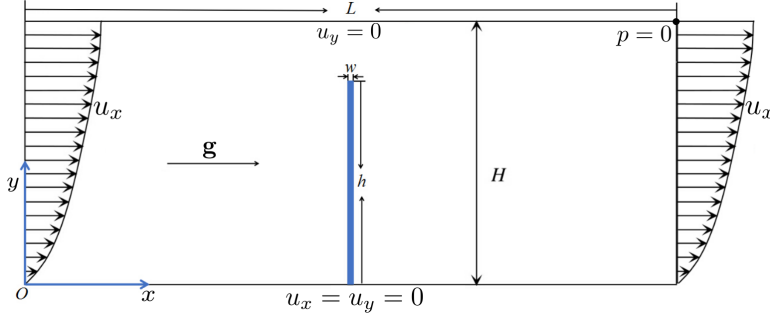


Figure 2: Computational domain and boundary conditions, taken from [16].

stated. The computational domain and boundary conditions are illustrated in Figure 2.

The inlet flow is in the x-direction and given by $u_x = 15.0y(2 - y) \sin(2\pi t)$. Gravity is not considered in the first test (i.e. $\mathbf{g} = 0$), and other fluid and solid properties are presented in Table 1.

Fluid	Leaflet
$L = 4.0 \text{ m}$	$w = 0.0212 \text{ m}$
$H = 1.0 \text{ m}$	$h = 0.8 \text{ m}$
$\rho^f = 100 \text{ kg/m}^3$	$\rho^s = 100 \text{ kg/m}^3$
$\mu^f = 10 \text{ N} \cdot \text{s/m}^2$	$\mu^s = 10^7 \text{ N/m}^2$

Table 1: Properties and domain size for test problem with a leaflet oriented across the flow direction.

The leaflet is approximated with 1200 linear triangles with 794 nodes (medium mesh size), and the corresponding fluid mesh is adaptive in the vicinity of the leaflet so that it has a similar size. A stable time step $\Delta t = 5.0 \times 10^{-4} \text{ s}$ is used in these initial simulations. The configuration of the leaflet is illustrated at different times in Figure 3.

Previously published numerical results are qualitatively similar to those in Figure 3 but show some quantitative variations. For example, [16] solved a fully-coupled system but the coupling is limited to a line, and the solid in their results (Figure 7 (l)) behaves as if it is slightly harder. Alternatively, [15] used a fractional step scheme to solve the FSI equations combined with a penalty method to enforce the incompressibility condition. In their results (Fig. 3 (h)) the leaflet behaves as if it is slightly softer than [16] and harder than [17]. In [17] a beam formulation is used to describe the solid. The fluid mesh is locally refined using hierarchical B-Splines, and the FSI equation is solved monolithically. The leaflet in their results (Fig. 34) behaves as softer than the other two considered here. Our results in Figure 3 are most similar to those of [17]. This may be seen more precisely by inspection of the graphs of the oscillatory motion of the leaflet tip in Figure 4 corresponding to Fig. 32 in [17]. We point out here that

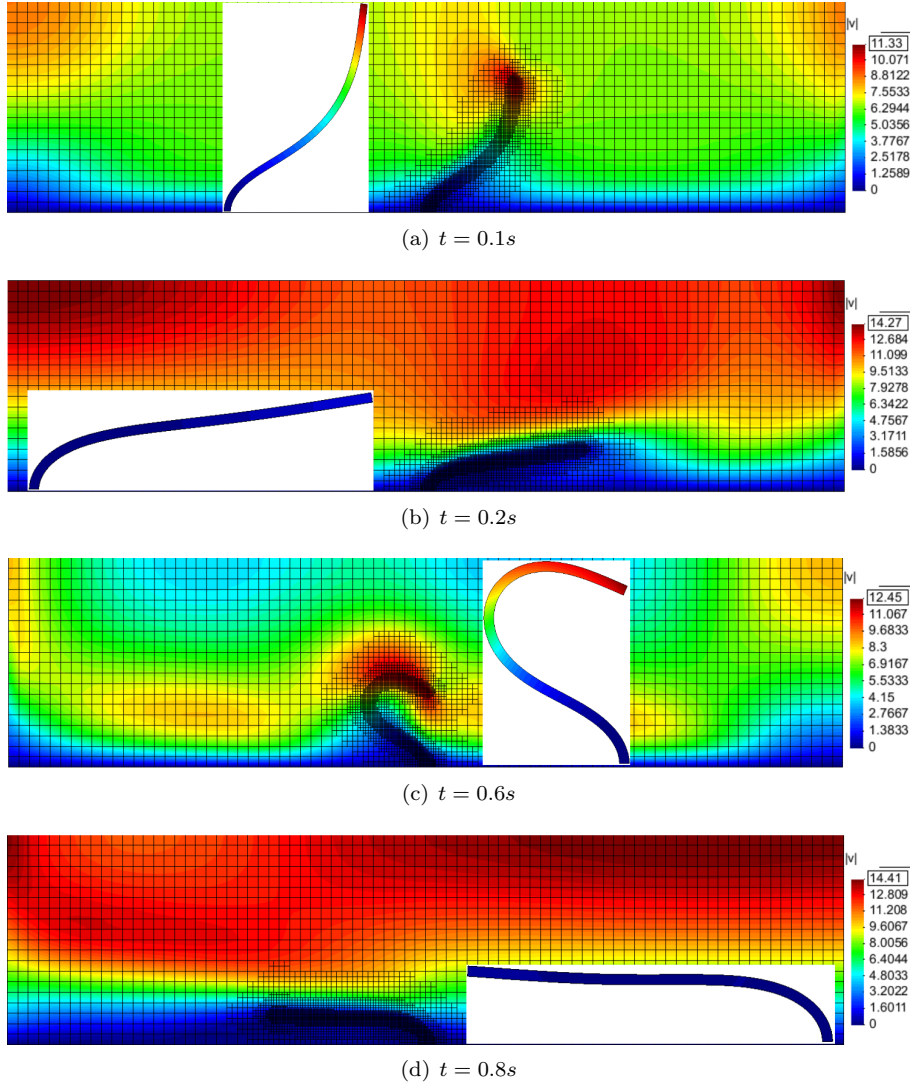


Figure 3: Configuration of leaflet and magnitude of velocity on the adaptive fluid mesh.

the explicit Taylor-Galerkin method is also used to solve the convection step for this test, and we gain almost the same accuracy using the same time step $\Delta t = 5.0 \times 10^{-4}s$.

Having validated our results for this example against the work of others, we shall use this test case to further explore more details of our method.

We commence by testing the influence of the ratio of fluid and solid mesh sizes $r_m = (\text{local fluid element area}) / (\text{solid element area})$. Fixing the fluid mesh size, three different solid mesh sizes are chosen: coarse (640 linear triangles with

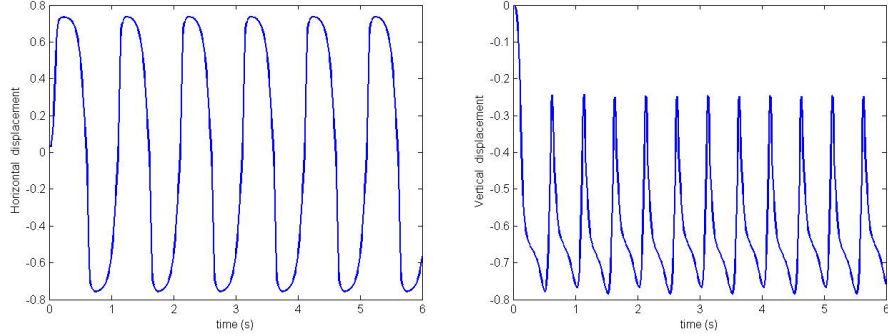


Figure 4: Evolution of horizontal and vertical displacement at top right corner of the leaflet.

403 nodes $r_m \approx 1.5$), medium (1200 linear triangles with 794 nodes $r_m \approx 3.0$)
and fine (2560 linear triangles with 1445 nodes $r_m \approx 5.0$), and a stable time
step $\Delta t = 5.0 \times 10^{-4} s$ is used. From these tests we observe that there is a slight
difference in the solid configuration for different meshes, as illustrated at $t = 0.6s$
in Figure 5, however the difference in displacement decreases as the solid mesh
becomes finer. Further, we found that $1.5 \leq r_m \leq 5.0$ ensures the stability of
the proposed UFEM approach. Note that we use a 9-node quadrilateral for the
fluid velocity and 3-node triangle for solid velocity, so $r_m \approx 3.0$ means the fluid
and solid mesh locally have a similar number of nodes for velocity.

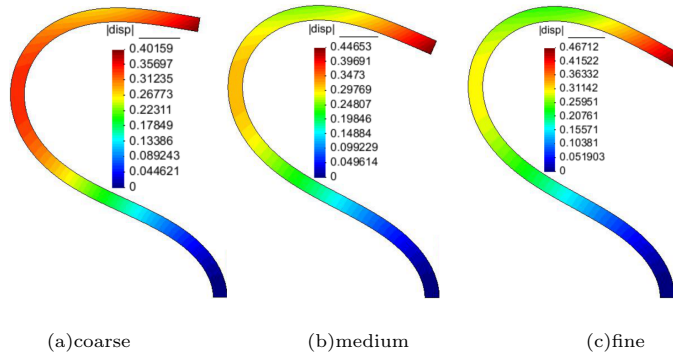


Figure 5: Configuration of leaflet for different mesh ratio r_m ,
and contour plots of displacement magnitude at $t = 0.6s$.

We next consider convergence tests undertaken for refinement of both the
fluid and solid meshes with the fixed ratio of mesh sizes $r_m \approx 3.0$. Four different
levels of meshes are used, the solid meshes are: coarse (584 linear triangles
with 386 nodes), medium (1200 linear triangles with 794 nodes), fine (2560
linear triangles with 1445 nodes), and very fine (3780 linear triangles with 2085
nodes). The fluid meshes have the corresponding sizes with the solid at their
maximum refinement level. As can be seen in Figure 6 and Table 2, the velocity

is converging as the mesh becomes finer.

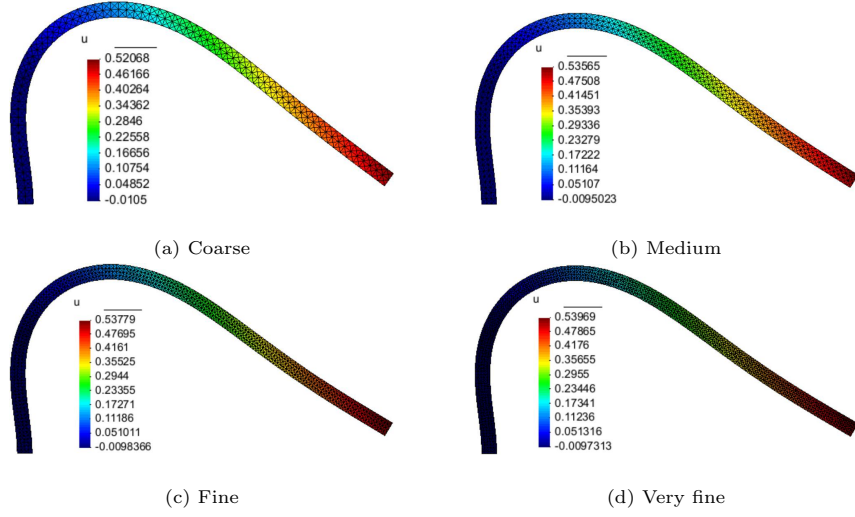


Figure 6: Contour plots of horizontal velocity at $t = 0.5s$.

Between different mesh sizes	Difference of maximum horizontal velocity at $t = 0.5s$
coarse and medium	0.01497
medium and fine	0.00214
fine and very fine	0.00190

Table 2: Comparison of maximum velocity for different meshes.

280 In addition, we consider tests of convergence in time using a fixed ratio of fluid and solid mesh sizes $r_m \approx 3.0$. Using the medium solid mesh size and the same fluid mesh size as above, results are shown in Figure 7 and Table 3. From these it can be seen that the velocities are converging as the time step decreases.

Steps sizes compared	Difference of maximum horizontal velocity at $t = 0.5s$
$\Delta t = 2.0 \times 10^{-3}$ and $\Delta t = 1.0 \times 10^{-3}$	0.00854
$\Delta t = 1.0 \times 10^{-3}$ and $\Delta t = 5.0 \times 10^{-4}$	0.00517
$\Delta t = 5.0 \times 10^{-4}$ and $\Delta t = 2.5 \times 10^{-4}$	0.00263

Table 3: Comparison of maximum velocity for different time step size.

285 Finally, in order to assess the robustness of our approach, we vary each of the physical parameters using three different cases as shown in Figure 8. A medium mesh size with fixed $r_m \approx 3.0$ is used to undertake all of these tests.

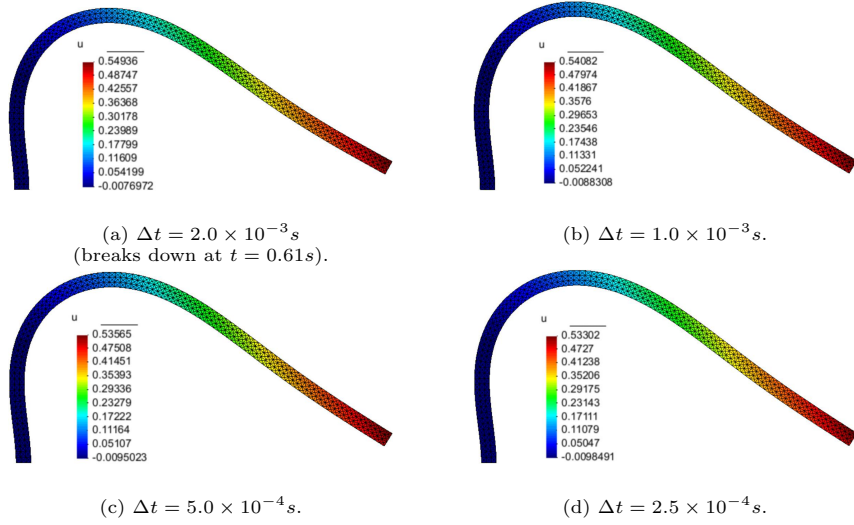


Figure 7: Contour plots of horizontal velocity at $t = 0.5s$.

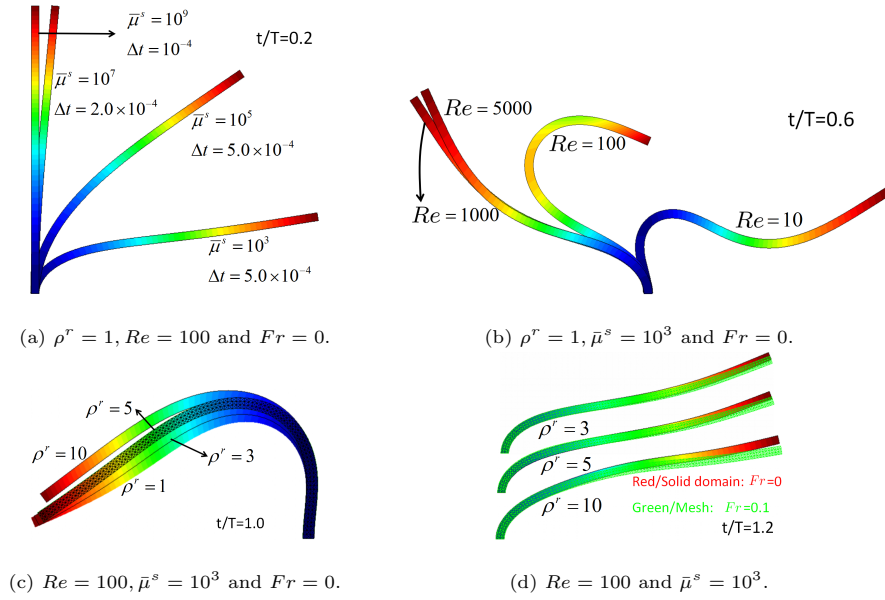


Figure 8: Parameters sets and results, $\Delta t = 5.0 \times 10^{-4} s$ for Group (b)~(d).

The dimensionless parameters shown in Figure 8 are defined as: $\rho^r = \frac{\rho^s}{\rho^f}$, $\bar{\mu}^s = \frac{\mu^s}{\rho^f U^2}$, $Re = \frac{\rho^f U H}{\mu^f}$ and $Fr = \frac{g H}{U^2}$ where the average velocity $U = 10$ in this example. $T = 1$ is the period of inlet flow.

It can be seen from the results of group (a) that the larger the value of

shear modulus $\bar{\mu}^s$ the harder the solid behaves, however a smaller time step is required. For the case of $\bar{\mu}^s = 10^9$, the solid behaves almost like a rigid body, as we would expect. From results of group (b), it is clear that the Reynolds Number (Re) has a large influence on the behavior of the solid. The density and gravity have relatively less influence on the behavior of solid in this problem which can be seen from the results of group (c) and group (d).

5.2. Oscillation of a flexible leaflet oriented along the flow direction

The following test problem that we consider is taken from [26], which describes an implementation on a ALE fitted mesh. It has since been used as a benchmark to validate different numerical schemes [17, 18]. The geometry and boundary conditions are shown in Figure 9.

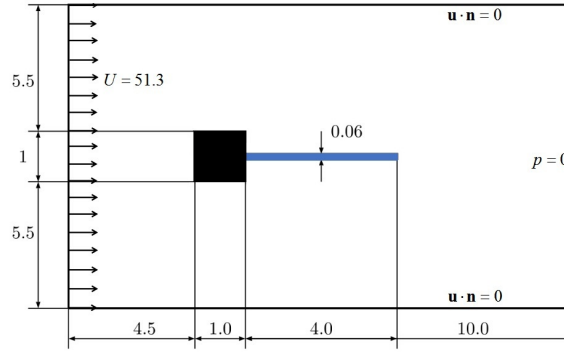


Figure 9: Computational domain and boundary condition for oscillation of flexible leaflet.

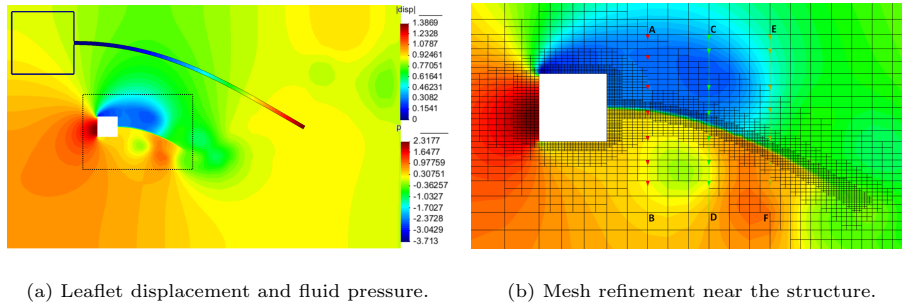


Figure 10: Contour plots of leaflet displacement and fluid pressure.

For the fluid, the viscosity and density are $\mu^f = 1.82 \times 10^{-4}$ and $\rho^f = 1.18 \times 10^{-3}$ respectively. For the solid, we use shear modulus $\mu^s = 9.2593 \times 10^{-5}$ and density $\rho^s = 0.1$. The leaflet is divided by 1063 3-node linear triangles with 666 nodes, and the corresponding fluid mesh locally has a similar node density to the leaflet ($r_m \approx 3.0$). First the Least-squares method is tested and a stable time step $\Delta t = 1.0 \times 10^{-3}s$ is used. A snapshot of the leaflet deformation

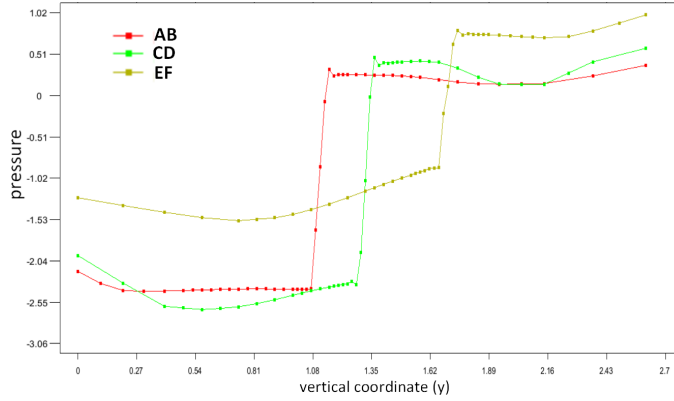


Figure 11: Distribution of pressure across the leaflet on the three lines in Figure 10 (b).

and fluid pressure at $t = 5.44s$ are illustrated in Figure 10. In Figure 11, the distributions of pressure across the leaflet corresponding to the three lines (AB, CD and EF) in Figure 10 (b) are plotted, from which we can observe that the sharp jumps of pressure across the leaflet are captured.

The evolution of the vertical displacement of the leaflet tip with respect to time is plotted in Figure 12(a). Both the magnitude (1.34) and the frequency (2.94) have a good agreement with the result of [26], using a fitted ALE mesh and of [17], using a monolithic unfitted mesh approach. The Taylor-Galerkin method is also tested which uses $\Delta t = 2.0 \times 10^{-4}s$ as a stable time step, and a corresponding result is shown in 12(b) which has a similar magnitude (1.24) and frequency (2.86). These results are all within the range of values in [17, Table 4]. Note that since the initial condition before oscillation for these simulations is an unstable equilibrium, the first perturbation from this regime is due to numerical disturbances. Consequently, the initial transient regimes observed for the two methods (implicit Least-squares and explicit Taylor-Galerkin methods) are quite different. It is possible that an explicit method causes these numerical perturbations more easily, therefore makes the leaflet start to oscillate at an earlier stage than when using the implicit Least-squares approach.

5.3. Solid disc in a cavity flow

This numerical example is used to compare our UFEM with the IFEM, which is cited in [11, 27]. In order to compare some details, we also implement the IFEM, but we implemented it on an adaptive mesh with hanging nodes, and we use the isoparametric FEM interpolation function rather than the discretized delta function or RKPM function of [9, 10].

The fluid's density and viscosity are 1 and 0.01 respectively, and the following solid properties are chosen to undertake the tests: $\rho^s=1$ and $\mu^s=0.1$ or 1. The horizontal velocity on the top boundary of the cavity is prescribed as 1 and the vertical velocity is fixed to be 0 as shown in Figure 13. The velocities on the

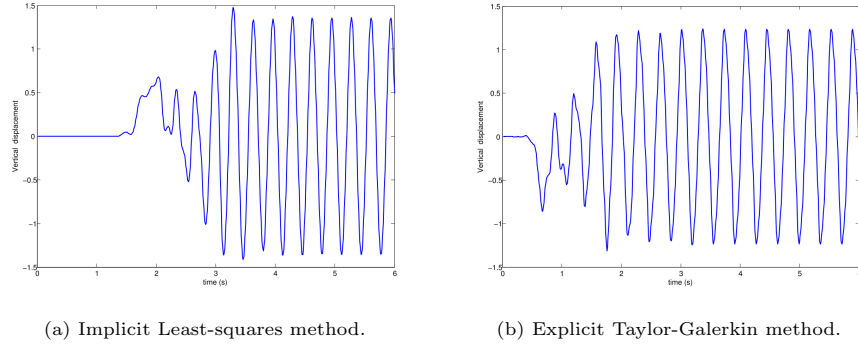


Figure 12: Displacement of leaflet tip as a function of time.

other three boundaries are all fixed to be 0, and pressure at the bottom-left point is fixed to be 0 as a reference point.

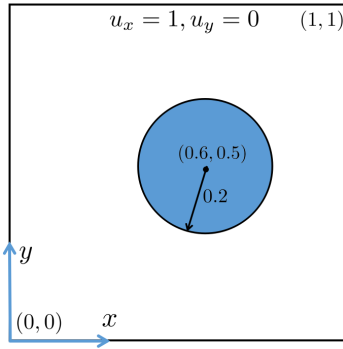


Figure 13: Computational domain for cavity flow, taken from [27].

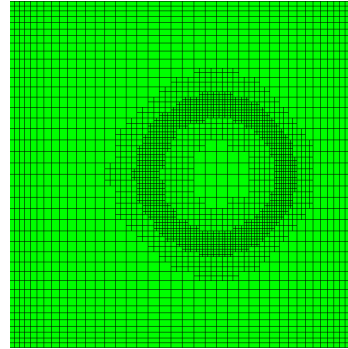


Figure 14: Adaptive mesh for cavity flow.

In order to compare the UFEM and IFEM, we use the same meshes for fluid and solid: the solid mesh has 2381 nodes and the fluid mesh locally has a similar number of nodes (adaptive, see Figure 14). First the implicit Least-squares method is used to solve the convection step, and the time step is $\Delta t = 1.0 \times 10^{-3}$. Figure 15 and Figure 16 show the configuration of the disc at different stages, from which we do not observe significant differences of the velocity norm even for a long run as shown in Figure 16 (b). Then the explicit Taylor-Galerkin method is tested, and we achieve almost the same accuracy by using the same time step. The magnitudes of velocity at the same stages of Figure 16 are presented in Figure 17.

We should mention that for the case $\mu^s = 0.1$, as the disc arrives at the top of the cavity (time > 3.0) the quality of the solid mesh does begin to deteriorate using our UFEM. We do not currently seek to improve the mesh quality (using an arbitrary Lagrangian-Eulerian (ALE) update [7], for example) however this would be necessary in order to reduce the shear modulus further

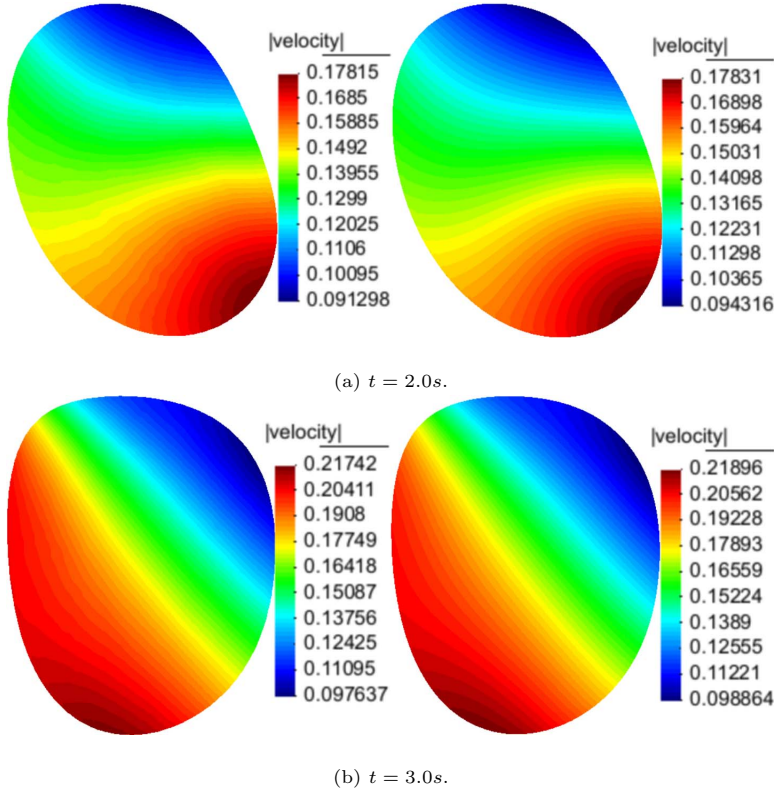


Figure 15: Velocity norm for a soft solid ($\mu^s = 0.1$) in a driven cavity flow using UFEM (left) and IFEM (right).

without compromising the quality of the solid mesh.

Conversely, a large μ^s makes the solid behave like a rigid body. For the
 355 proposed UFEM, we can use $\mu^s = 100$ or larger without changing the time step,
 whereas for the IFEM the simulation always breaks down for $\mu^s = 100$, however
 small the time step, due to the huge FSI force on the right-hand side of the FSI
 system.

5.4. Solids in a channel with gravity

360 We first simulate a falling disc due to gravity in order to further validate the
 accuracy of the UFEM. We then show a simulation of the evolution of different
 shapes of solids falling and rising in a channel in order to show the flexibility
 and robustness of the proposed UFEM.

The test of a falling disc in a channel is cited by [10, 18] in order to validate
 365 the IFEM and a monolithic method respectively. The computational domain
 and parameters are illustrated in Figure 18 and Table 4 respectively. The fluid
 velocity is fixed to be 0 on all boundaries except the top one.

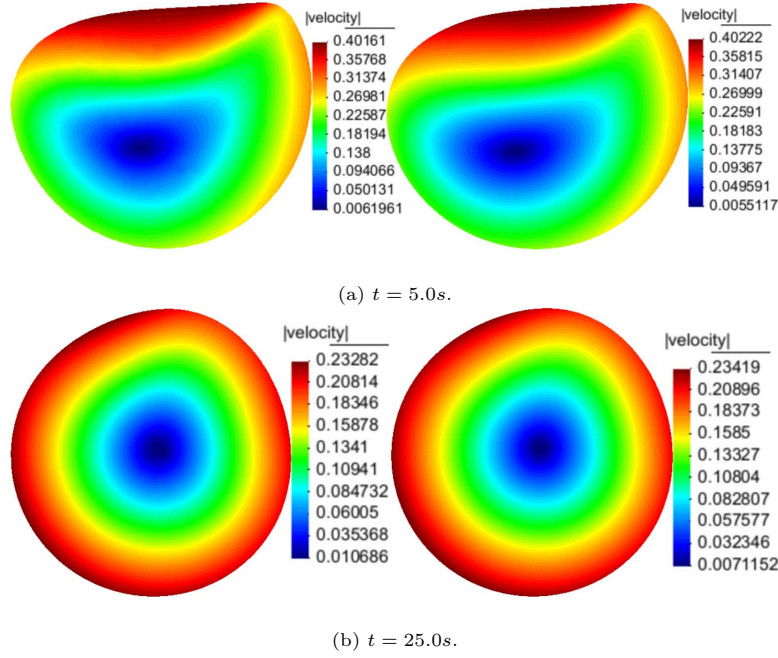


Figure 16: Velocity norm for a soft solid ($\mu^s = 1.0$) in a driven cavity flow using UFEM (left) and IFEM (right), Least-squares method for convection step.

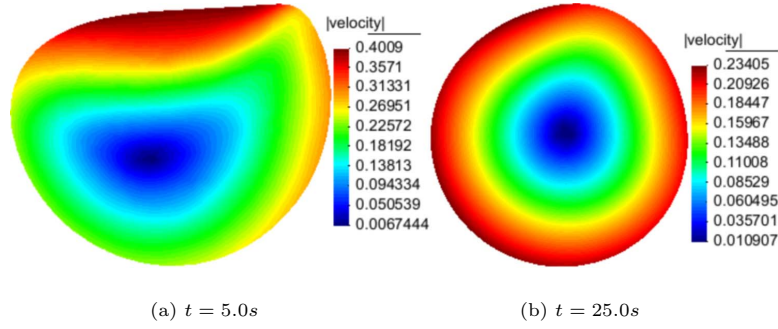


Figure 17: Velocity norm for $\mu^s = 1.0$, Taylor-Galerkin method for convection step.

There is also an empirical solution of a rigid ball falling in a viscous fluid [18], for which the terminal velocity, u_t , under gravity is given by

$$u_t = \frac{(\rho^s - \rho^f) g r^2}{4\mu^f} \left(\ln \left(\frac{L}{r} \right) - 0.9157 + 1.7244 \left(\frac{r}{L} \right)^2 - 1.7302 \left(\frac{r}{L} \right)^4 \right), \quad (64)$$

where ρ^s and ρ^f are the density of solid and fluid respectively, μ^f is viscosity of the fluid, $g = 980 \text{ cm/s}^2$ is acceleration due to gravity, $L = W/2$ and r is the radius of the falling ball. We choose $\mu^s = 10^8 \text{ dyne/cm}^2$ to simulate a

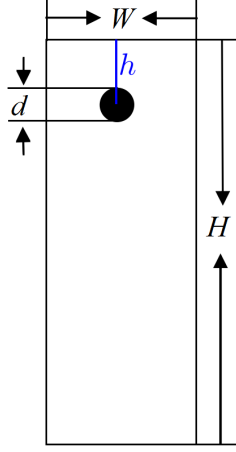


Figure 18: Computational domain for a falling disc.

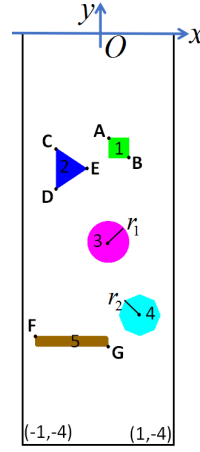


Figure 19: Computational domain for different shapes of solids with different properties.

Fluid	Disc
$W = 2.0 \text{ cm}$	$d = 0.0125 \text{ cm}$
$H = 4.0 \text{ cm}$	$h = 0.5 \text{ cm}$
$\rho^f = 1.0 \text{ g/cm}^3$	$\rho^s = 1.2 \text{ g/cm}^3$
$\mu^f = 1.0 \text{ dyne} \cdot \text{s/cm}^2$	$\mu^s = 10^8 \text{ dyne/cm}^2$
$g = 980 \text{ cm/s}^2$	$g = 980 \text{ cm/s}^2$

Table 4: Fluid and material properties of a falling disc.

rigid body here, and $\mu^s = 10^{12} \text{ dyne/cm}^2$ is also applied, which gives virtually identical result.

Three different meshes are used: the disc boundary is represented with 28 nodes (coarse), 48 nodes (medium), or 80 nodes (fine). The fluid mesh near the solid boundary has the same mesh size, and a stable time step $t = 0.005s$ is used for all the three cases. The Least-squares method is used to treat the convection step in all these tests. A local snapshot of the vertical velocity with the adaptive mesh is shown in Figure 20. From the fluid velocity pattern around the disc, we can observe that the disc behaves like a rigid body as expected. In addition, the evolution of the velocity of the mid-point of the disc is shown in Figure 21, from which it can be seen that the numerical solution converges from below to the empirical solution.

Reference [18] uses a monolithic method to simulate multiple rigid and deformable discs in a gravity channel. We have implemented this example and obtain very similar results. Rather than replicate these here however, we instead show a more complex example, as illustrated in Figure 19. The computational domain, boundary conditions and the fluid properties are the same as the above one-disc test. All the solids are numbered at their initial positions as

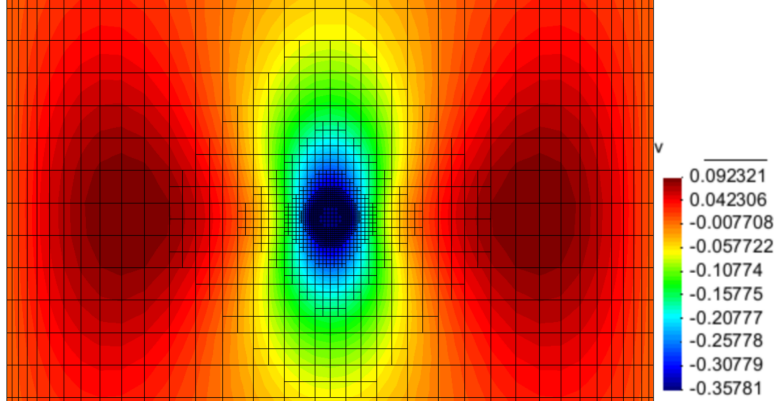


Figure 20: Contour of vertical velocity at $t = 1s$ (fine mesh).

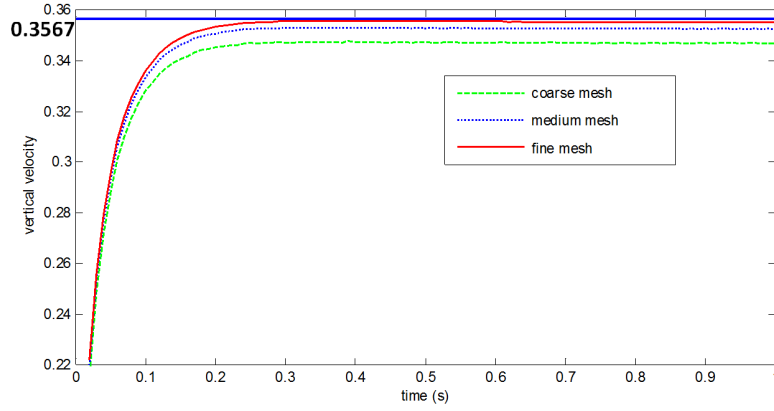


Figure 21: Evolution of velocity at the center of a falling disc.
(The blue solid line represents the empirical solution from formula (64).)

shown in Figure 19 with $A(0, -1)$, $B(0.2, -1.2)$, $C(-0.5, -1.1)$, $D(-0.5, -1.5)$,
 390 $E(-0.2, -1.3)$, $F(-0.7, -2.9)$ and $G(0, -3)$. The center and radius (r_1) of the
 3^{rd} solid (circle) are $(0, -2)$ and 0.2 respectively, and the center and radius (r_2)
 the 4^{th} solid (octagon) are $(0.3, -2.7)$ and 0.2 respectively. The solid properties
 are illustrated in Table 5.

A high resolution of each solid boundary is used in this simulation as shown
 395 in Figure 22 (a), which can guarantee the mesh quality during the whole process
 of evolution, and a stable time step $t = 0.002s$ is used. Snapshots of the solids
 at different times are shown in Figure 22 and 23.

6. Conclusion and future works

In this article we introduce a new unified finite element method (UFEM) for
 400 fluid-structure interaction, which can be applied to a wide range of problems,

No. of solid	Density (g/cm^3)	Shear modulus ($dyne/cm^2$)
1	1.3	10^4
2	1.2	10^3
3	1.0	10
4	0.8	10^6
5	0.7	10^2

Table 5: Properties for multi-solids falling in a channel as shown in Figure 19.

from small deformation to very large deformation and from very soft solids through to very rigid solids. Several numerical examples, which are widely used in the literature of Immersed FEM and monolithic methods, are implemented to validate the proposed UFEM.

405 The UFEM combines features from the IFEM and from monolithic methods. Nevertheless, it differs from each of them in the following aspects. Firstly, UFEM is a semi-explicit (explicitly linearizing the constitutive equation of solid and implicitly coupling FSI interaction) scheme, similar to IFEM, however UFEM solves the solid equations and fluid equations together while the classical
410 IFEM does not solve the solid equations; secondly, both UFEM and monolithic methods solve solid equations, however UFEM solves one velocity field in the solid domain using FEM interpolation, while monolithic methods solve one velocity field and one displacement field in the solid domain using Lagrangian multipliers. In summary therefore we believe that UFEM has the potential to
415 offer the robustness and range of operation of monolithic methods, but at a computational cost that is much closer to that of the immersed finite element methods.

The following generalizations of our proposed UFEM approach will be considered in the future: (1) Implementation in 3D using adaptive mesh with hang-
420 ing nodes; (2) implementation for non-Newtonian flow; (3) an efficient preconditioned iterative solver for the UFEM algebraic system; (4) a second order splitting scheme in time.

Appendix A. A method to treat hanging nodes

An adaptive mesh with hanging nodes reduces the number of degrees of freedom compared to uniform refinement, hence, decreases the cost of computation.
425 However, the nature of hanging nodes has the potential to cause discontinuity and breaks the framework of the finite element shape functions, which, therefore, needs special treatment in finite element codes.

In order to treat the hanging nodes, one can construct a conforming shape function [28, 29] or constrain and cancel the degree of freedom at the hanging
430 nodes [29, 30]. The former is very appealing and leads to optimal convergence, but it is difficult to extend to high-order shape functions [31]. In this article we will adopt the latter method and only use 2-level hanging nodes, which means at most 2 hanging nodes are allowed in one element (this can be guaranteed by

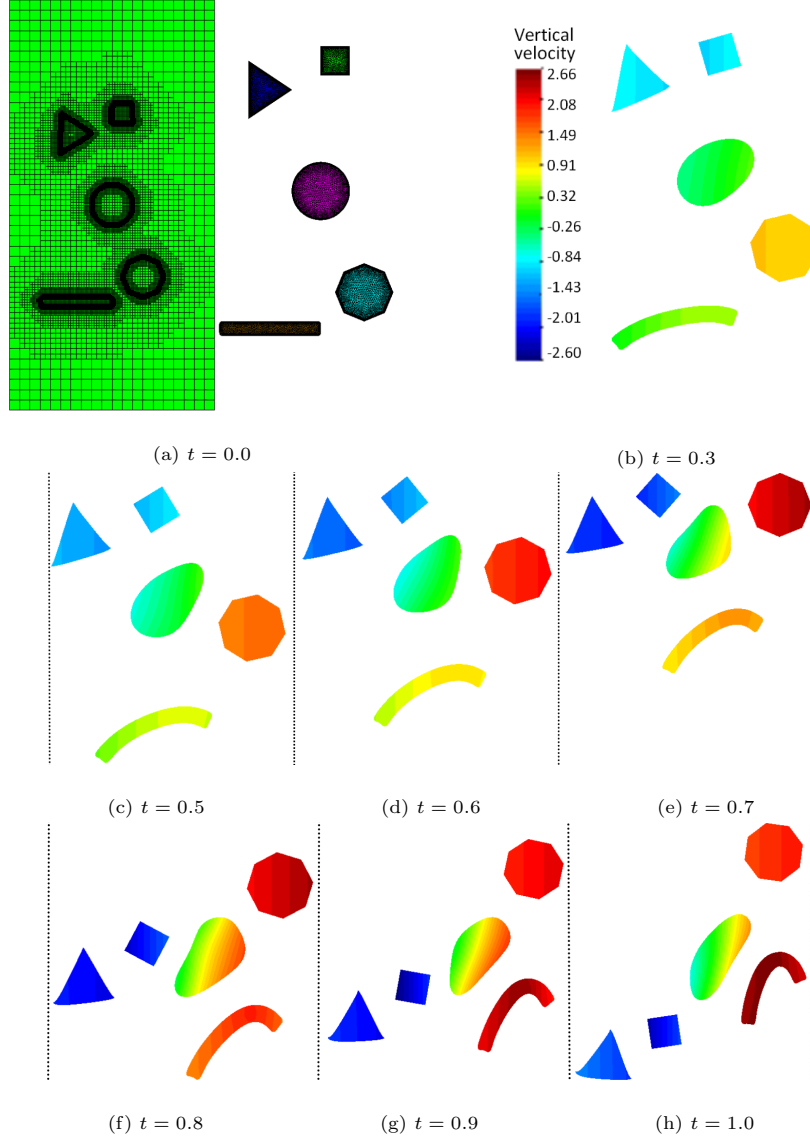


Figure 22: Contours of vertical velocity at different times.

435 imposing safety layers to ensure that neighbouring element nodes differ by more
 than one level of refinement). The implementation of arbitrary-level hanging
 nodes can be found in [31, 32, 33].

For a quadrilateral element, when the velocity is interpolated by biquadratic
 shape functions and the pressure is interpolated by bilinear shape functions, the
 440 implementation of hanging nodes must be different for each, as shown in Figure

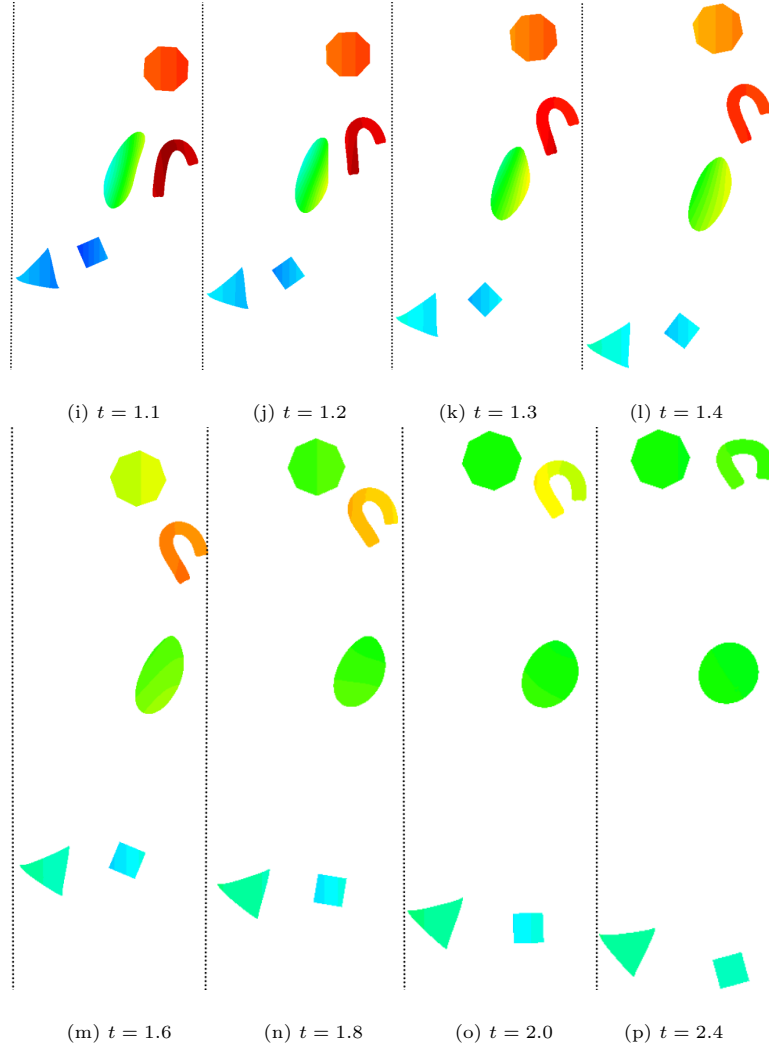


Figure 23: Contours of vertical velocity at different times (continued).

A.1.

For example, when velocity is interpolated, point D is a hanging node for element II, and point E is a hanging node for element III. When pressure is interpolated, point C is a hanging node for both the element II and III. Take
445 element II for example, if we use the constraint method to cancel the hanging nodes degree of freedom, then

$$u_i^D = \frac{3}{8}u_i^A - \frac{1}{8}u_i^B + \frac{3}{4}u_i^C \quad i = (1, 2) \quad (\text{A.1})$$

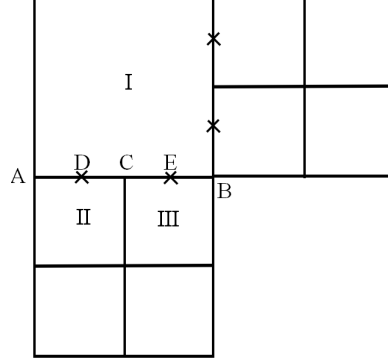


Figure A.1: Elements with hanging nodes.

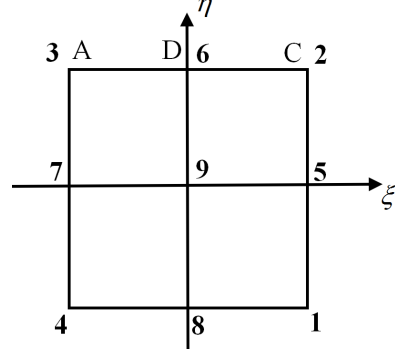


Figure A.2: Element II in Figure A.1 in the reference coordinate system.

$$p^C = \frac{1}{2}p^A + \frac{1}{2}p^B \quad (\text{A.2})$$

where u_i and p are velocity components and pressure respectively defined at the corresponding nodes. The interpolation coefficients can be calculated by putting edge AB in a one dimensional finite element reference coordinate system.

450 Notice that when computing the element matrix II, point B is outside of the element, but the element matrix II still contributes to node B because of the hanging node D . So we can treat the two points, B and D , as a master-slave couple, which means letting them share the same equation number in the final global linear equation system. However one should modify the element matrix
455 II according to (A.1) and (A.2) in the following way before assembling it to the global matrix.

Suppose the element II is enumerated in the reference coordinate system as shown in Figure A.2. Then, formulae (A.1) and (A.2) imply the following equations:

$$\begin{pmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \\ u_i^4 \\ u_i^5 \\ u_i^6 \\ u_i^7 \\ u_i^8 \\ u_i^9 \end{pmatrix} = \mathbf{D}^v \begin{pmatrix} u_i^1 \\ u_i^2 \\ u_i^3 \\ u_i^4 \\ u_i^5 \\ u_i^B \\ u_i^7 \\ u_i^8 \\ u_i^9 \end{pmatrix}, \quad \mathbf{D}^v = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & \frac{3}{4} & \frac{3}{8} & & & -\frac{1}{8} & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix}. \quad (\text{A.3})$$

$$\begin{pmatrix} p^1 \\ p^2 \\ p^3 \\ p^4 \end{pmatrix} = \mathbf{D}^p \begin{pmatrix} p^1 \\ p^B \\ p^3 \\ p^4 \end{pmatrix}, \quad \mathbf{D}^p = \begin{bmatrix} 1 & & & \\ & \frac{1}{2} & \frac{1}{2} & \\ & & 1 & \\ & & & 1 \end{bmatrix}. \quad (\text{A.4})$$

One should use matrices \mathbf{D}^v and \mathbf{D}^p to modify the element matrix II. Suppose \mathbf{K}_e is the stiffness matrix of element II without consideration of hanging nodes, and the unknowns are arranged in the following column vector.

$$(u_1^1, u_1^2, \dots, u_1^9, v_1^1, v_1^2 \dots v_1^9, p^1, p^2 \dots p^4)^T. \quad (\text{A.5})$$

460 It is clear that $\mathbf{K}_e = [k_{ij}]$ is a $n \times n$ ($n=22$) matrix, and it could be modified by the following pseudocode, which distribute the contribution of hanging nodes to the corresponding nodes according to formula (A.1).

for j=1 to n $k_{i_1j} = k_{i_1j} + k_{i_0j} \cdot 3/8$ $k_{i_2j} = k_{i_2j} + k_{i_0j} \cdot 3/4$ end	for j=1 to n $k_{ji_1} = k_{ji_1} + k_{ji_0} \cdot 3/8$ $k_{ji_2} = k_{ji_2} + k_{ji_0} \cdot 3/4$ end	for j=1 to n $k_{i_0j} = -k_{i_0j}/8$ $k_{ji_0} = -k_{ji_0}/8$ end
---	---	---

465 Let $i_0 = 6$, $i_1 = 3$, and $i_2 = 2$ (based on (A.5)), sequentially executing the above three pieces of codes would modify the matrix \mathbf{K}_e corresponding to the first component of velocity, and let $i_0 = 15$, $i_1 = 12$, and $i_2 = 11$ (based on (A.5)), executing the above codes would modify the matrix \mathbf{K}_e corresponding to the second component of velocity. Similarly, in order to modify the matrix corresponding to pressure, one can execute the following codes which are based on formula (A.2):

for j=1 to n $k_{i_1j} = k_{i_1j} + k_{i_2j}/2$ end	for j=1 to n $k_{ji_1} = k_{ji_1} + k_{ji_2}/2$ end	for j=1 to n $k_{i_2j} = -k_{i_2j}/2$; $k_{ji_2} = -k_{ji_2}/2$ end
---	---	--

470 where $i_1 = 21$ and $i_2 = 20$ based on (A.5). Executing all the above pieces of codes is equivalent to performing the following matrix multiplication.

$$\begin{bmatrix} \mathbf{D}^v & & \\ & \mathbf{D}^v & \\ & & \mathbf{D}^p \end{bmatrix}^T \mathbf{K}_e \begin{bmatrix} \mathbf{D}^v & & \\ & \mathbf{D}^v & \\ & & \mathbf{D}^p \end{bmatrix}. \quad (\text{A.6})$$

475 The modification of the mass matrix is similar but easier if a lumped mass is adopted, though it is unnecessary to present details here. Once the element matrix is modified, it can then be assembled directly to the global matrix and therefore implement the constraint of the hanging nodes, because the hanging node shares the same equation number with its related node in the neighbouring element.

References

References

- 480 [1] G. Hou, J. Wang, A. Layton, Numerical methods for fluid-structure interaction: a review, *Commun. Comput. Phys* 12 (2) (2012) 337–377. doi:10.4208/cicp.291210.290411s.
- [2] U. Küttler, W. A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, *Computational Mechanics* 43 (1) (2008) 61–72. doi:10.1007/s00466-008-0255-5.
- 485 [3] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction, *Computers & Structures* 87 (11-12) (2009) 793–801. doi:10.1016/j.compstruc.2008.11.013.
- 490 [4] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems, *Computer Methods in Applied Mechanics and Engineering* 193 (1-2) (2004) 1–23. doi:10.1016/j.cma.2003.09.006.
- [5] M. Heil, A. L. Hazel, J. Boyle, Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches, *Computational Mechanics* 43 (1) (2008) 91–101. doi:10.1007/s00466-008-0270-6.
- 495 [6] R. L. Muddle, M. Mihajlović, M. Heil, An efficient preconditioner for monolithically-coupled large-displacement fluid–structure interaction problems with pseudo-solid mesh updates, *Journal of Computational Physics* 231 (21) (2012) 7315–7334. doi:10.1016/j.jcp.2012.07.001.
- 500 [7] R. C. Peterson, P. K. Jimack, M. A. Kelmanson, The solution of two-dimensional free-surface problems using automatic mesh generation, *International journal for numerical methods in fluids* 31 (6) (1999) 937–960. doi:10.1002/(SICI)1097-0363(19991130)31:6<937::AID-FLD906>3.0.CO;2-p.
- 505 [8] M. A. Walkley, P. H. Gaskell, P. K. Jimack, M. A. Kelmanson, J. L. Summers, Finite element simulation of three-dimensional free-surface flow problems, *J Sci Comput* 24 (2) (2005) 147–162. doi:10.1007/s10915-004-4611-0.
- 510 [9] L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu, Immersed finite element method, *Computer Methods in Applied Mechanics and Engineering* 193 (21) (2004) 2051–2067. doi:10.1016/j.cma.2003.12.044.
- [10] L. Zhang, M. Gay, Immersed finite element method for fluid-structure interactions, *Journal of Fluids and Structures* 23 (6) (2007) 839–857. doi:10.1016/j.jfluidstructs.2007.01.001.
- 515

- [11] X. Wang, L. T. Zhang, Interpolation functions in the immersed boundary and finite element methods, *Computational Mechanics* 45 (4) (2009) 321–334. doi:10.1007/s00466-009-0449-5.
- 520 [12] X. Wang, C. Wang, L. T. Zhang, Semi-implicit formulation of the immersed finite element method, *Computational Mechanics* 49 (4) (2011) 421–430. doi:10.1007/s00466-011-0652-z.
- [13] X. Wang, L. T. Zhang, Modified immersed finite element method for fully-coupled fluid–structure interactions, *Computer Methods in Applied Mechanics and Engineering* 267 (2013) 150–169. doi:10.1016/j.cma.2013.07.019.
- 525 [14] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Périaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, *Journal of Computational Physics* 169 (2) (2001) 363–426. doi:10.1006/jcph.2000.6542.
- 530 [15] Z. Yu, A DLM/FD method for fluid/flexible-body interactions, *Journal of Computational Physics* 207 (1) (2005) 1–27. doi:10.1016/j.jcp.2004.12.026.
- [16] F. P. Baaijens, A fictitious domain/mortar element method for fluid-structure interaction, *International Journal for Numerical Methods in Fluids* 35 (7) (2001) 743–761. doi:10.1002/1097-0363(20010415)35:7<743::AID-FLD109>3.0.CO;2-A.
- 535 [17] C. Kadapa, W. Dettmer, D. Perić, A fictitious domain/distributed lagrange multiplier based fluid–structure interaction scheme with hierarchical b-spline grids, *Computer Methods in Applied Mechanics and Engineering* 301 (2016) 1–27. doi:10.1016/j.cma.2015.12.023.
- 540 [18] C. Hesch, A. Gil, A. A. Carreño, J. Bonet, P. Betsch, A mortar approach for fluid–structure interaction problems: Immersed strategies for deformable and rigid bodies, *Computer Methods in Applied Mechanics and Engineering* 278 (2014) 853–882. doi:10.1016/j.cma.2014.06.004.
- 545 [19] C. S. Peskin, The immersed boundary method, *Acta numerica* 11 (2002) 479–517. doi:10.1016/j.cma.2015.12.023.
- [20] A. Robinson-Mosher, C. Schroeder, R. Fedkiw, A symmetric positive definite formulation for monolithic fluid structure interaction, *Journal of Computational Physics* 230 (4) (2011) 1547–1566. doi:10.1016/j.jcp.2010.11.021.
- 550 [21] E. Hachem, S. Feghali, R. Codina, T. Coupez, Anisotropic adaptive meshing and monolithic variational multiscale method for fluid–structure interaction, *Computers & Structures* 122 (2013) 88–100. doi:10.1016/j.compstruc.2012.12.004.

- 555 [22] E. Bodewig, M. Abramowitz, Matrix calculus, Elsevier BV, 2014.
- [23] D. Boffi, L. Gastaldi, A fictitious domain approach with lagrange multiplier for fluid-structure interactions, Numer. Math. doi:10.1007/s00211-016-0814-1.
- [24] O. Zienkiewicz, The finite element method for fluid dynamics, 6th Edition, Elsevier BV, 2005.
- 560 [25] P. B. Bochev, M. D. Gunzburger, Least-squares finite element methods, Vol. 166, Springer Science & Business Media, 2009.
- [26] W. A. Wall, Fluid-struktur-interaktion mit stabilisierten finiten elementen, Ph.D. thesis, Universitt Stuttgart (1999). doi:10.18419/OPUS-127.
- 565 [27] H. Zhao, J. B. Freund, R. D. Moser, A fixed-mesh method for incompressible flow-structure systems with finite solid deformations, Journal of Computational Physics 227 (6) (2008) 3114–3140. doi:10.1016/j.jcp.2007.11.019.
- [28] A. K. Gupta, A finite element for transition from a fine to a coarse grid, International Journal for Numerical Methods in Engineering 12 (1) (1978) 35–45. doi:10.1002/nme.1620120104.
- 570 [29] T.-P. Fries, A. Byfut, A. Alizada, K. W. Cheng, A. Schrder, Hanging nodes and XFEM, International Journal for Numerical Methods in Engineering 86 (4-5) (2010) 404–430. doi:10.1002/nme.3024.
- 575 [30] W. Bangerth, O. Kayser-Herold, Data structures and requirements for hp finite element software, ACM Transactions on Mathematical Software 36 (1) (2009) 1–31. doi:10.1145/1486525.1486529.
- [31] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, E. Rank, Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes, Computational Mechanics 55 (3) (2015) 499–517. doi:10.1007/s00466-014-1118-x.
- 580 [32] P. Šolín, J. Červený, I. Doležel, Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM, Mathematics and Computers in Simulation 77 (1) (2008) 117–132. doi:10.1016/j.matcom.2007.02.011.
- 585 [33] E. Ooi, H. Man, S. Natarajan, C. Song, Adaptation of quadtree meshes in the scaled boundary finite element method for crack propagation modelling, Engineering Fracture Mechanics 144 (2015) 101–117. doi:10.1016/j.engfracmech.2015.06.083.